



Modul Praktikum

Fisika Komputasi II

Dengan Matlab GUI

**IGA Widagda
Fisika FMIPA UNUD
2014**

Kata Pengantar

Sebelumnya kami memanjatkan puji syukur kepada Tuhan Yang Maha Esa, karena berkat rahmat-Nya maka kami bisa menerbitkan *Modul Praktikum Fisika Komputasi II dengan Matlab Gui*. Kami berharap semoga buku ini dapat membantu mahasiswa Fisika FMIPA UNUD dalam menguasai materi kuliah Fisika Komputasi II.

Kami juga menghaturkan terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan modul ini :

- Bapak Ir. S. Poniman, M.Si., Ketua Jurusan Fisika FMIPA UNUD
- Teman – teman dosen dan karyawan Jurusan Fisika, FMIPA, UNUD
- Istriku, Kusuma dewi dan kedua anakku, Widya dan Lestari

Kami menyadari bahwa modul praktikum ini masih banyak kekurangannya, untuk itu kami sangat mengharapkan kritik dan saran dari semua pihak untuk kesempurnaan modul ini.

PENYUSUN

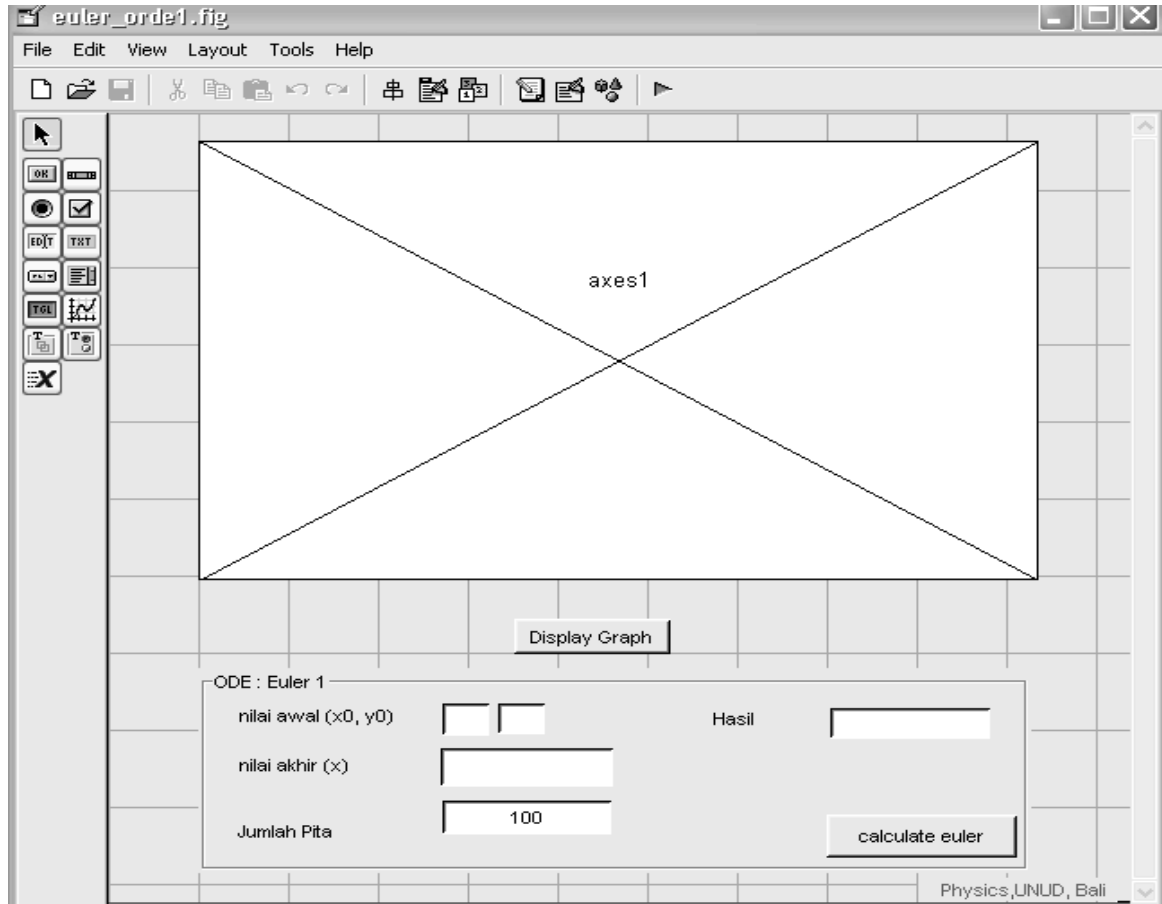
DAFTAR ISI

Kata Pengantar	i
Daftar Isi	ii
4 Penyelesaian Persamaan Diferensial	30
5 Penyelesaian Persamaan Linear	62
6 Interpolasi	66
7 Regresi	70
Daftar Pustaka	

IV Persamaan Differensial

4.1 Program Komputer Euler orde -1

4.1.1.Rancangan/ desain GUI



Gambar 4.1 Desain GUI – Euler orde-1

4.1.2. Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Pushbutton1	String	Display Graph
		Tag	show_graph_pushbutton
3	Static text1	String	nilai awal (x0,y0)
4	Static text2	String	nilai akhir (x)
5	Static text3	String	Jumlah pita
6	Static text4	String	Hasil
7	Edit text1	String	

		Tag	nilai_awal_x0_edit
8	Edit text2	String	
		Tag	nilai_awal_y0_edit
9	Edit text3	String	
		Tag	nilai_akhir_edit
10	Edit text4	String	100
		Tag	jml_pita_edit
10	Edit text5	String	
		Tag	Hasil_euler_edit
11	Panel1	Title	ODE : Euler 1
12	Pushbutton1	String	Calculate Euler
		Tag	calculate_euler_pushbutton

4.1.3. Kode Program (*source code*)

4.1.3.1 Function : *show_graph_pushbutton_Callback*

```
function show_graph_pushbutton_Callback(hObject, eventdata, handles)
axes(handles.axes1);
handles.x=0:0.1:100
handles.y=4*handles.x+5;
plot(handles.x,handles.y);
title('Pers.Diff dy/dx = 4x+5');
xlabel('x');
ylabel('y');
guidata(hObject,handles);
```

4.1.3.2 Function : *calculate_euler_pushbutton_Callback*

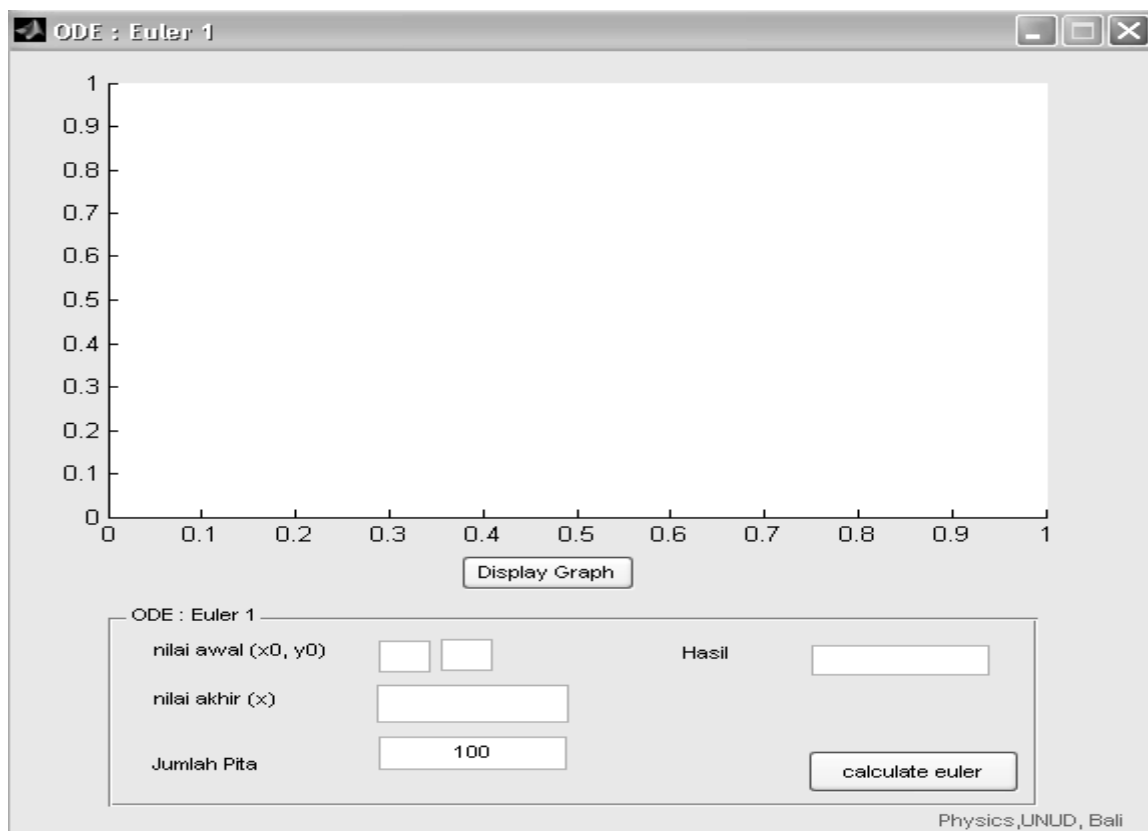
```
function calculate_euler_pushbutton_Callback(hObject, eventdata, handles)
x1 = str2num(get(handles.nilai_awal_x0_edit,'String'));
y1 = str2num(get(handles.nilai_awal_y0_edit,'String'));
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
pita = str2num(get(handles.jml_pita_edit,'String'));
n=0;
delt=100;
eps=1e-3;
while (abs(delt)>eps) %iterasi selama delt lebih besar dari eps
    n=n+1;
    %pita=pow(2,n); %2^n
    delx=(x2-x1)/pita;
    x=x1;
```

```

y=y1;
while (x<x2)
    %Pers.Diferensial yang dicari solusinya
    %*****
    %fx=y;
    fx=4*x+5;
    %fx=4;
    %*****
    y=y+fx*delx; %iterasi Euler orde-1
    x=x+delx;
end
if (n>1)
    delt=y-y2;
end
y2=y;
end
set(handles.hasil_euler_edit,'String',num2str(y));
    
```

4.1.4 Hasil Eksekusi (RUN) program

Hasil eksekusi program Euler 1 dapat dilihat dalam gambar 4.2.

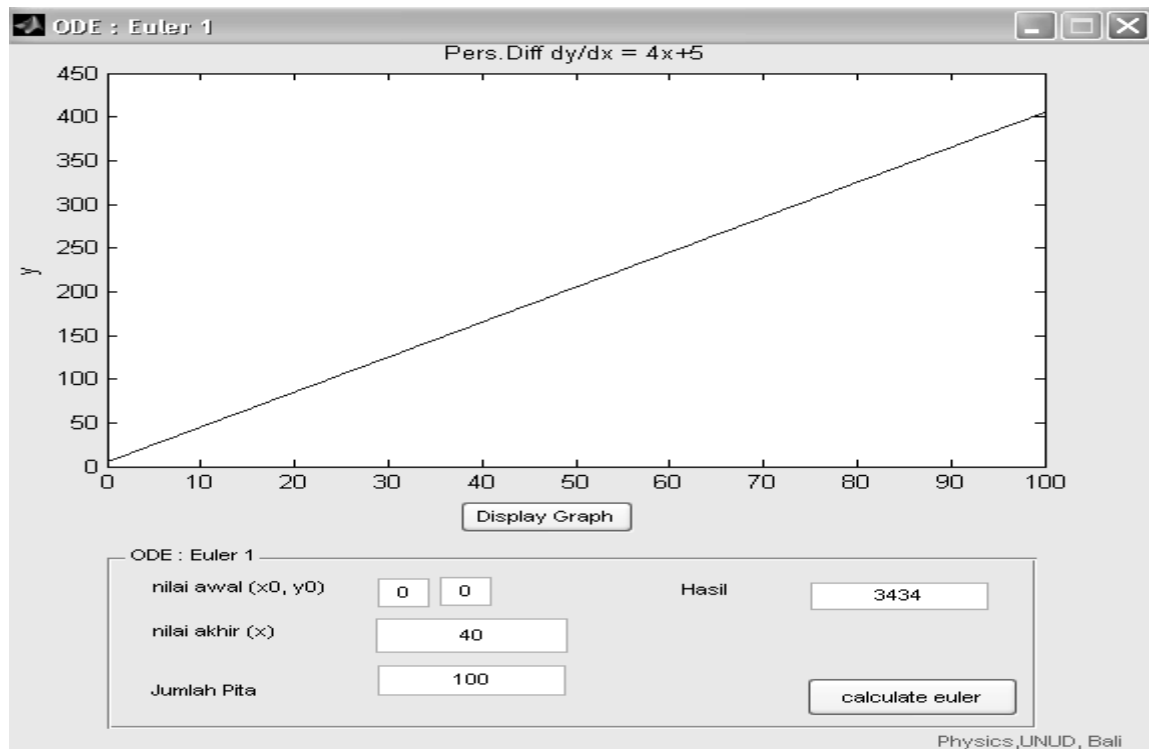


Gambar 4.2 Hasil Eksekusi program Euler 1

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal (x_0, y_0) misal : 0 0
- Masukkan nilai akhir (x) misal : 10
- Masukkan jumlah pita misal : 100
- Klik tombol *calculate euler*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.3.



Gambar 4.3 Hasil Akhir eksekusi program Euler 1

Penjelasan program :

-persamaan diferensial yang akan dicari solusinya adalah :

$$dy/dx = 4x + 5$$

-nilai awal (x_0, y_0) : 0 0

Artinya $x_0 = 0$ dan $y_0 = 0$

-nilai akhir (x) : 40

Artinya akan dihitung solusi persamaan diferensial saat $x = 40$ atau $y(40)$

-jumlah pita : 100

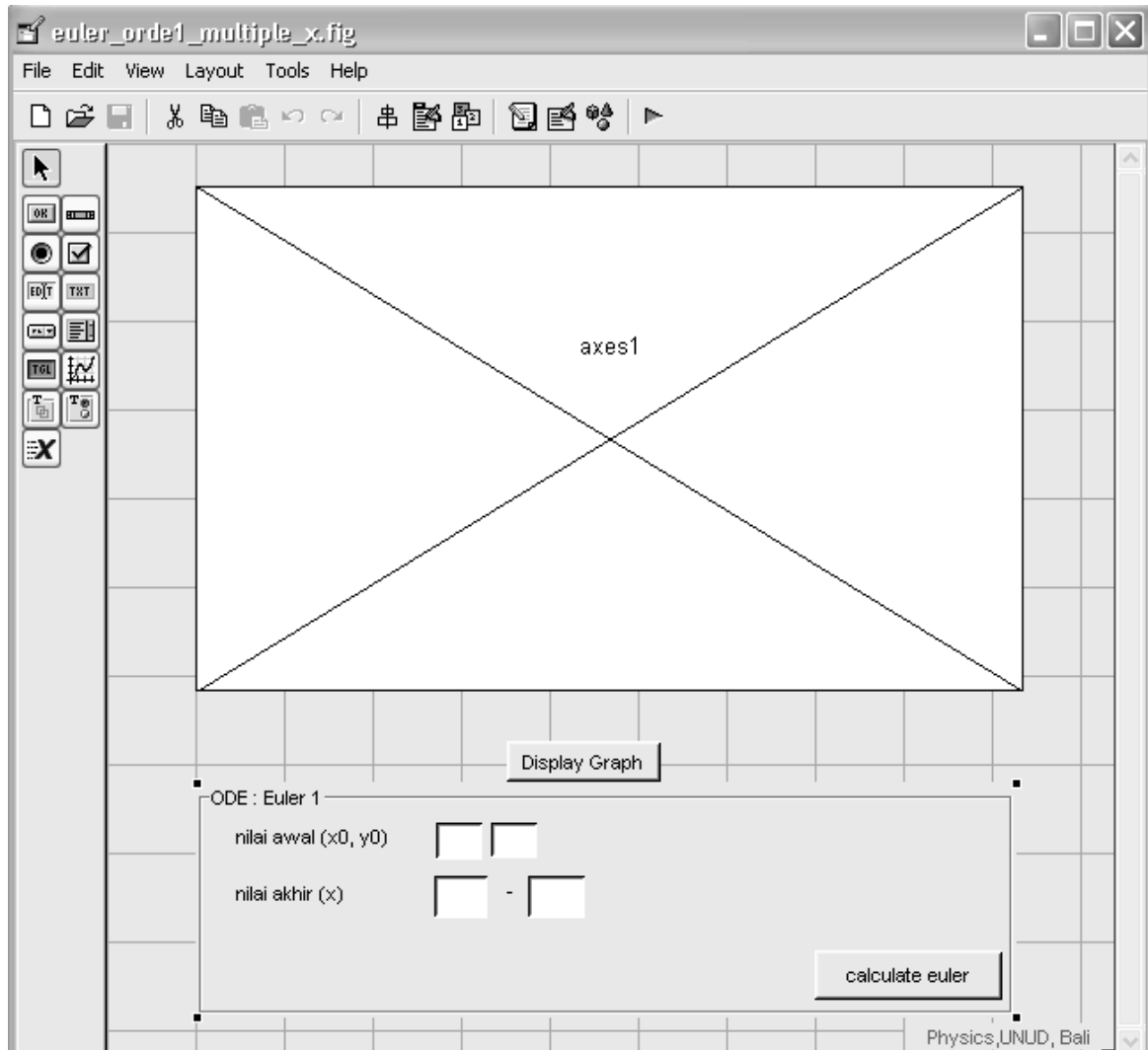
Antara x_0 dan x akan dibagi menjadi 100 segmen atau pita

-hasil : 3434

Solusi persamaan diferensial $y(40) = 3434$

4.2 Program Komputer Euler orde -1 (multiple x) – kasus : $dy/dx = 4x + 5$

4.2.1.Rancangan/ desain GUI



Gambar 4.4 Desain GUI – Euler orde-1 (multiple x)

4.2.2 Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Pushbutton1	String	Display Graph
		Tag	show_graph_pushbutton
3	Static text1	String	nilai awal (x0,y0)
4	Static text2	String	nilai akhir (x)
5	Edit text1	String	
		Tag	Nilai_awal_x0_edit
6	Edit text2	String	
		Tag	Nilai_awal_y0_edit

7	Edit text3	String	
		Tag	Nilai_akhir_edit
8	Edit text4	String	
		Tag	nilai_akhir2_edit
9	Panel1	Title	ODE : Euler 1
12	Pushbutton1	String	Calculate Euler
		Tag	calculate_euler_pushbutton

4.2.3. Kode Program (*source code*)

4.2.3.1 Function : *show_graph_pushbutton_Callback*

```
function show_graph_pushbutton_Callback(hObject, eventdata, handles)
axes(handles.axes1);
handles.x=0:0.1:100
handles.y=4*handles.x+5;
plot(handles.x,handles.y);
title('Pers.Diff dy/dx = 4x+5');
xlabel('x');
ylabel('y');
guidata(hObject,handles);
```

4.2.3.2 Function : *calculate_euler_pushbutton_Callback*

```
function calculate_euler_pushbutton_Callback(hObject, eventdata, handles)
x1 = str2num(get(handles.nilai_awal_xo_edit,'String'));
y1 = str2num(get(handles.nilai_awal_y0_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:1:x3; %x target from x2 to x3 with interval 1
for i=1:1:length(x_final)
    pita=10; %the number of segments / bands
    delx=(x2-x1)/pita; %the width of each segment /band
    x=x1;
    y_it(i)=y1;
    while (x<x_final(i))
        %Pers.Diferensial yang dicari solusinya
        %*****
        %fx=y;
        fx=4*x+5;
        %fx=4;
```

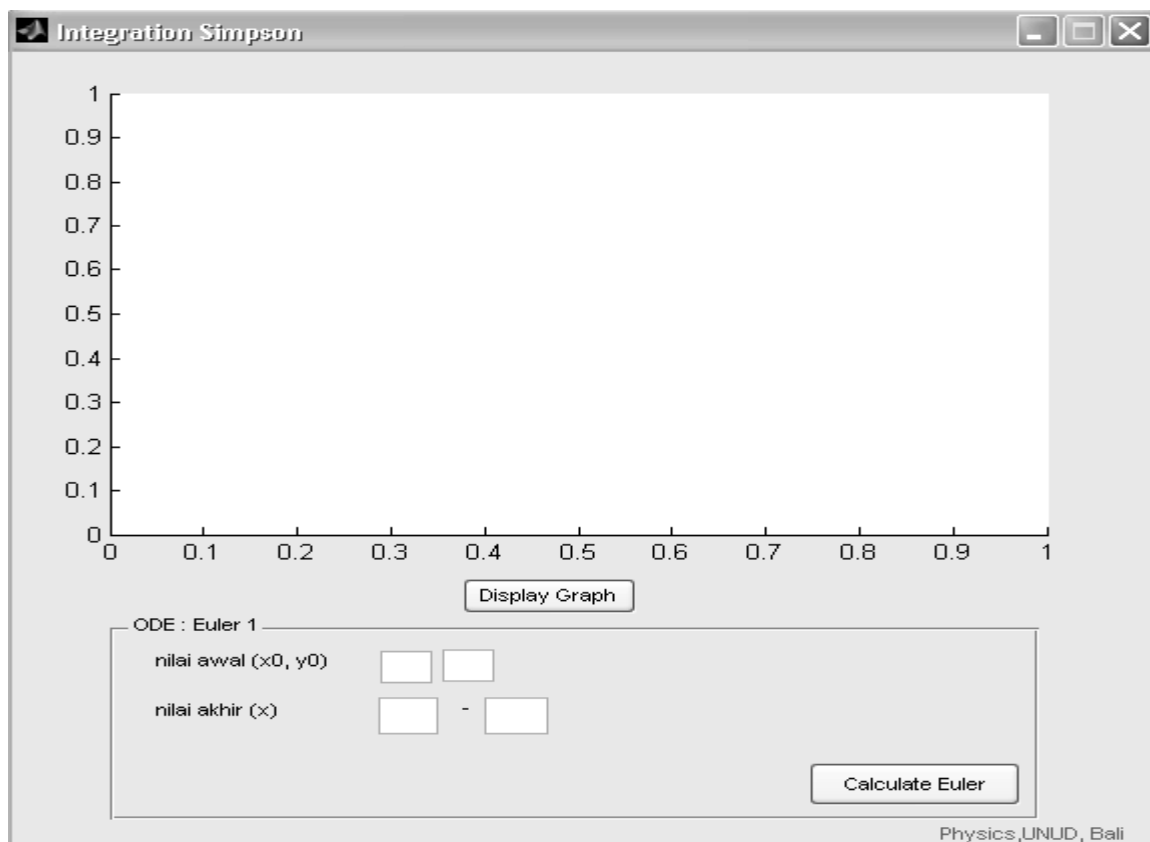
```

%*****
y_it(i)=y_it(i)+fx*delx; %iterasi Euler orde-1
x=x+delx;
end
end
cla;
axes(handles.axes1);
%display dy/dx
plot(handles.x,handles.y);
%text(10,10,'y');
title('Pers.Diff dy/dx = 4x+5');
xlabel('x');
ylabel('dy/dx');
%display y, the solution of dy/dx
plot(x_final,y_it,'r');
legend('dy/dx','y');
hold on;

```

4.2.4 Hasil Eksekusi (RUN) program

Hasil eksekusi program Euler 1 (multiple x) dapat dilihat dalam gambar 4.5.

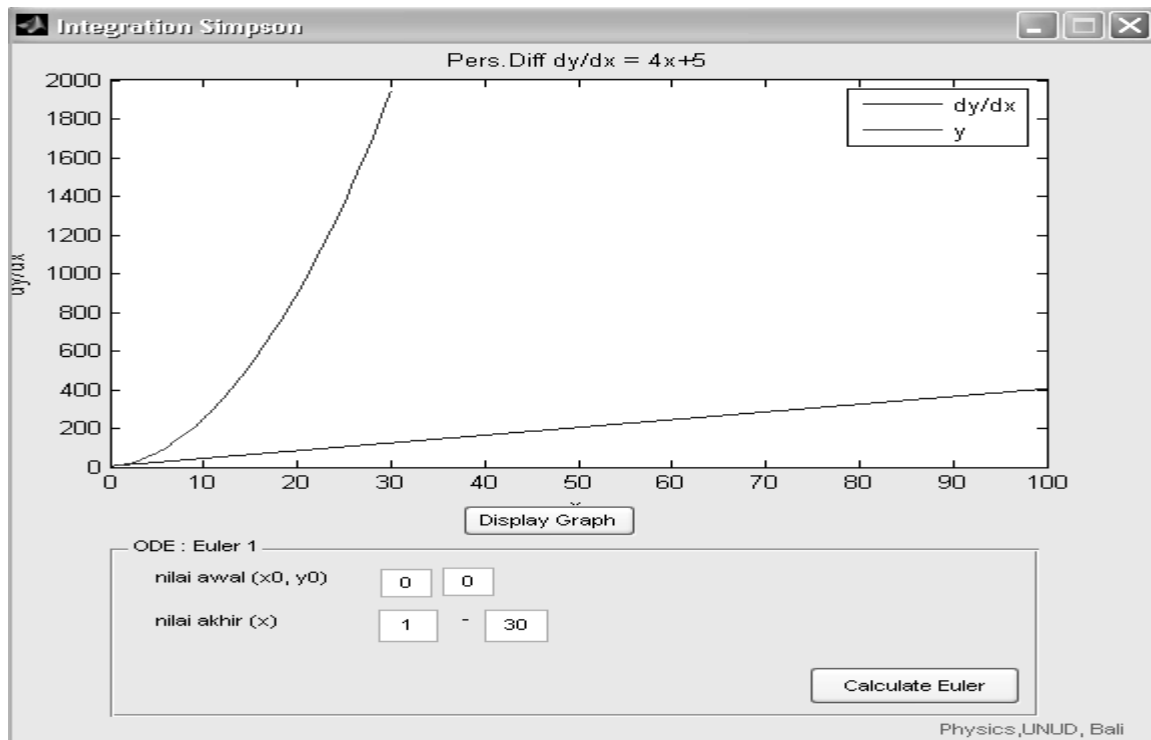


Gambar 4.5 Hasil Eksekusi program Euler 1 (multiple x)

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal (x_0, y_0) misal : 0 0
- Masukkan nilai akhir (x) misal : 10
- Masukkan jumlah pita misal : 100
- Klik tombol *calculate euler*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.6.



Gambar 4.6 Hasil Akhir eksekusi program Euler 1 (multiple x)

Penjelasan program :

-persamaan diferensial yang akan dicari solusinya adalah :

$$dy/dx = 4x + 5$$

-nilai awal (x_0, y_0) : 0 0

Artinya $x_0 = 0$ dan $y_0 = 0$

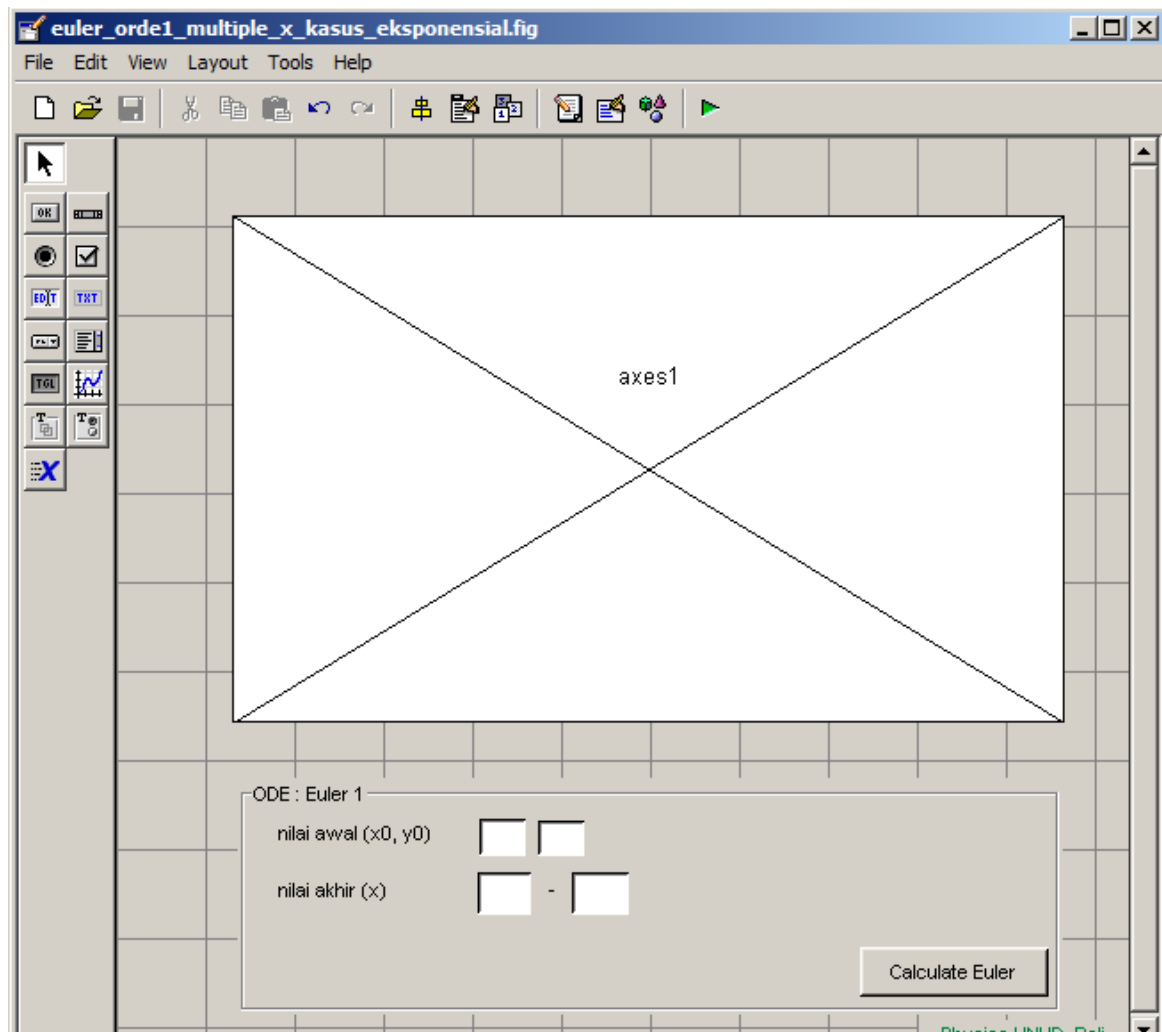
-nilai akhir (x) : 1 30

Artinya akan dihitung solusi persamaan diferensial saat $x = 1$ sampai $x=30$ atau $y(1)$ sampai $y(30)$

- hasil solusinya adalah 2 buah kurva yaitu kurva $dy/dx=4x+5$ dan kurva solusinya y dari $x=1$ sampai $x=30$

4.3 Program Komputer Euler orde -1 (multiple x) – kasus : $dy/dx = y$

4.3.1. Rancangan/ desain GUI



Gambar 4.7 Desain GUI – Euler orde-1 - kasus : $dy/dx = y$

4.3.2. Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Static text1	String	nilai awal (x0,y0)
3	Static text2	String	nilai akhir (x)
4	Edit text1	String	
		Tag	Nilai_awal_x0_edit
5	Edit text2	String	
		Tag	Nilai_awal_y0_edit
6	Edit text3	String	
		Tag	Nilai_akhir_edit
7	Edit text4	String	
		Tag	nilai_akhir2_edit

8	Panel1	Title	ODE : Euler 1
9	Pushbutton1	String	Calculate Euler
		Tag	calculate_euler_pushbutton

4.3.3. Kode Program (*source code*)

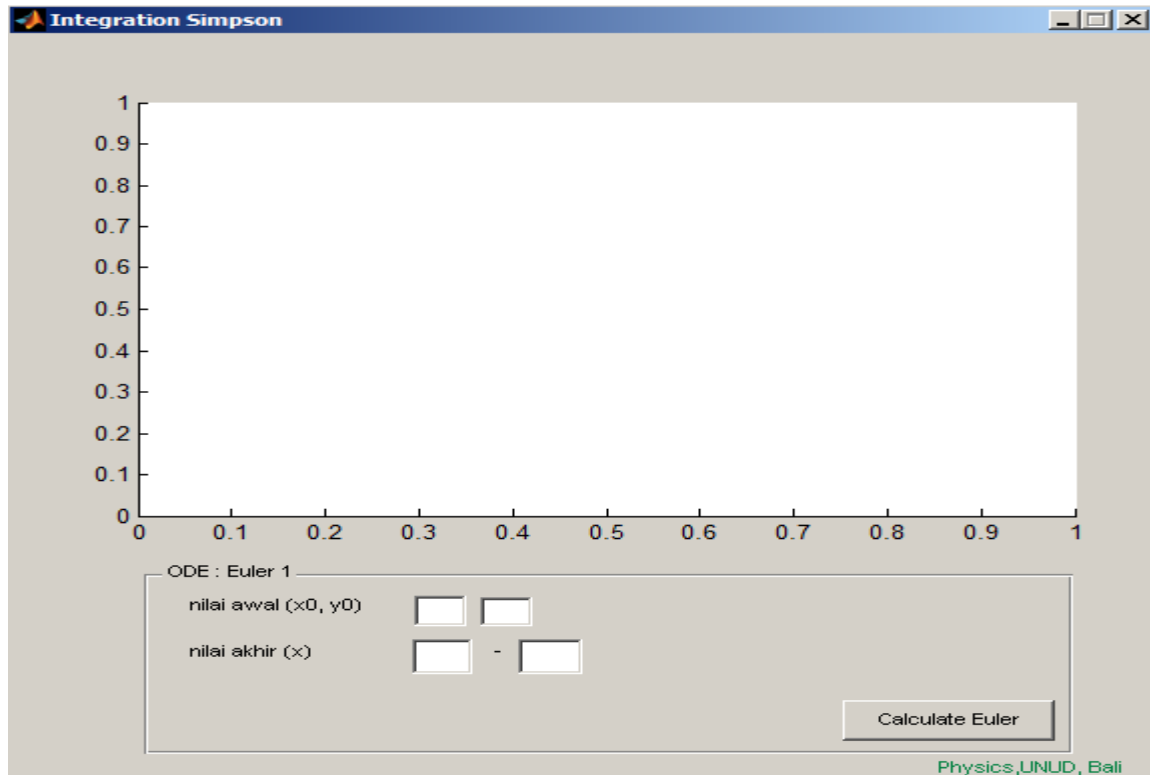
```

Function : calculate_euler_pushbutton_Callback
x1 = str2num(get(handles.nilai_awal_xo_edit,'String'));
y1 = str2num(get(handles.nilai_awal_y0_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:0.1:x3; %x target from x2 to x3 with interval 1
for i=1:1:length(x_final)
    pita=10; %the number of segments / bands
    delx=(x2-x1)/pita; %the width of each segment /band
    x=x1;
    y_it(i)=y1;
    while (x<x_final(i))
        %Pers.Differensial yang dicari solusinya
        %*****
        fx=y_it(i); %kasus : dy/dx = y
        %fx=4*x+5;
        %*****
        y_it(i)=y_it(i)+fx*delx; %iterasi Euler orde-1
        x=x+delx;
    end
end
cla;
axes(handles.axes1);
%display y, the solution of dy/dx
plot(x_final,y_it,'b');
title('Pers.Diff dy/dx = y');
xlabel('x');
ylabel('y');
hold on;

```

4.3.4 Hasil Eksekusi (*RUN*) program

Hasil eksekusi program Euler 1 (multiple x) dapat dilihat dalam gambar 4.8.

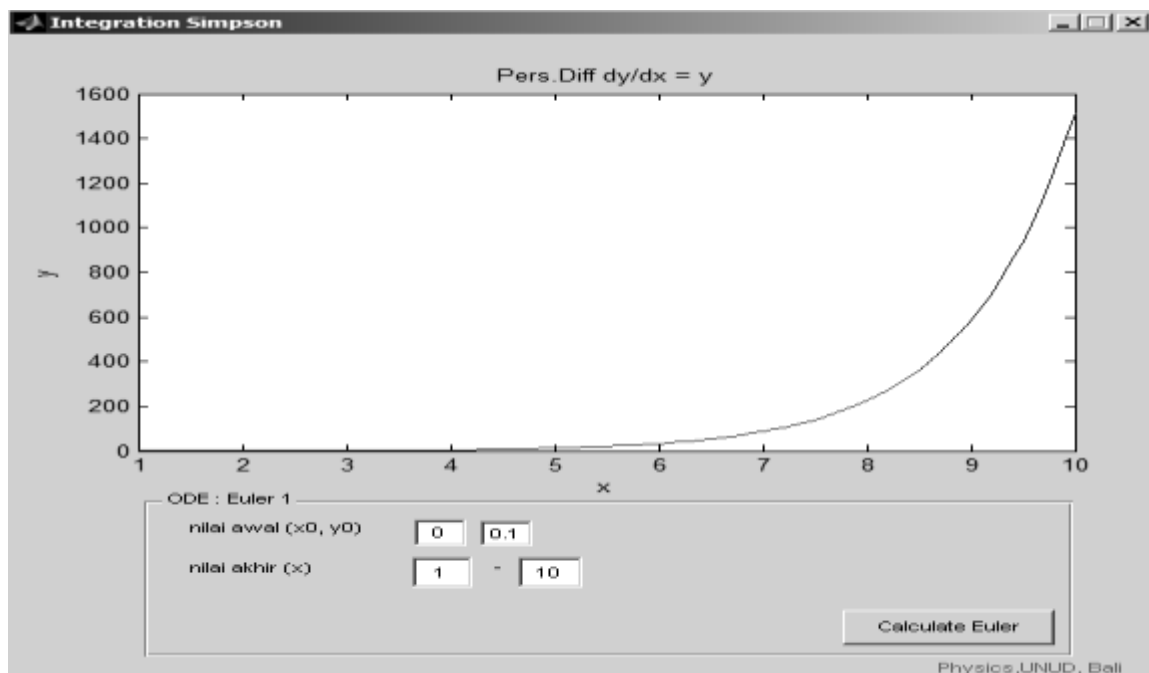


Gambar 4.8 Hasil Euler 1 (multiple x)-kasus : $dy/dx = y$

Cara kerja program :

- Masukkan nilai awal (x0,y0) misal : 0 0.1
- Masukkan nilai akhir (x) misal : 1 - 10
- Klik tombol *calculate euler*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.9.



Gambar 4.9 Hasil Akhir Euler – orde 1 – kasus : $dy/dx = y$

Penjelasan program :

-persamaan diferensial yang akan dicari solusinya adalah :

$$dy/dx = y$$

-nilai awal (x_0, y_0) : 0 0.1

Artinya $x_0 = 1$ dan $y_0 = 0.1$

-nilai akhir (x) : 1 10

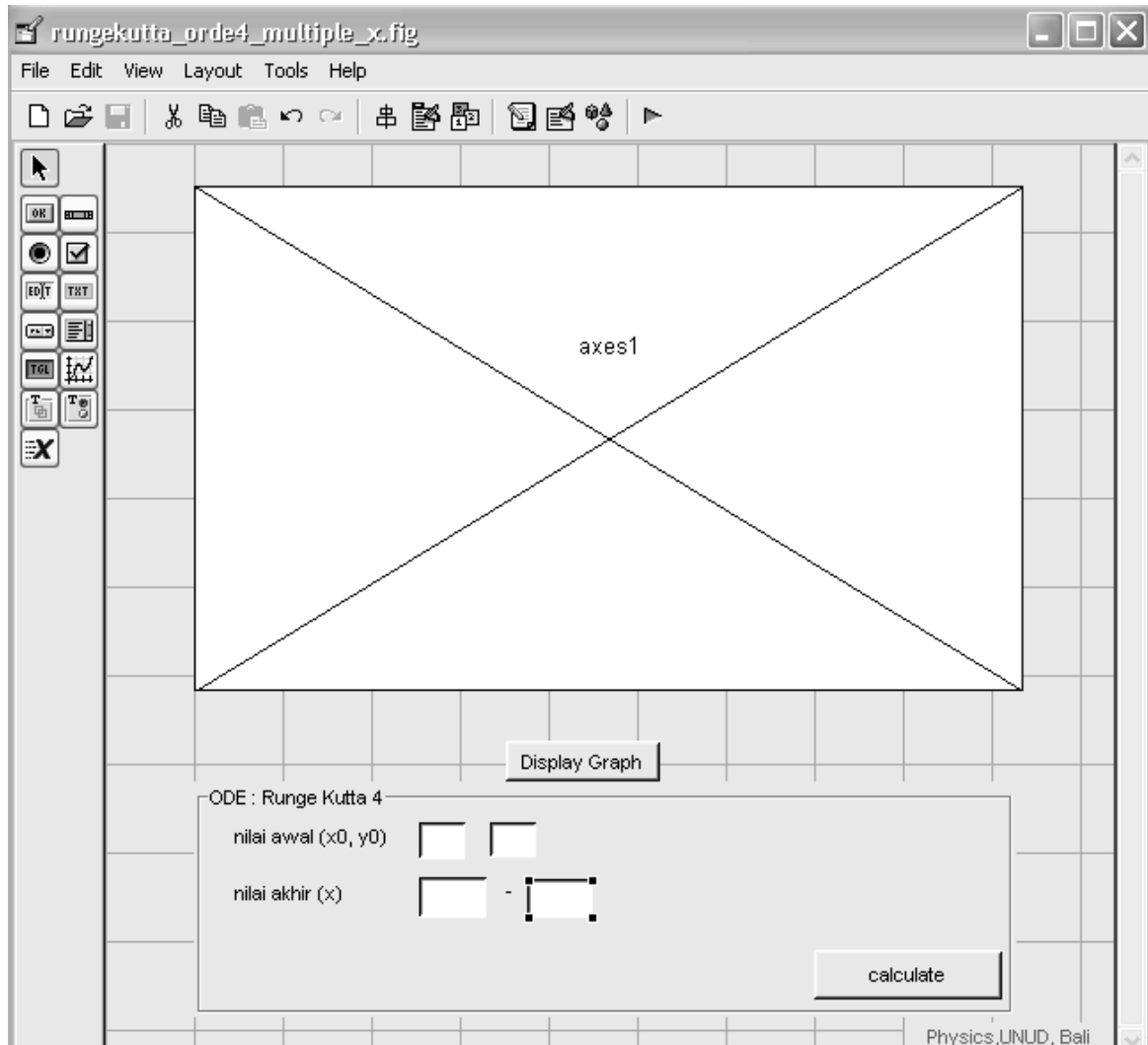
Artinya akan dihitung solusi persamaan diferensial saat $x = 1$ sampai $x=10$

atau $y(1)$ sampai $y(10)$

- hasil solusinya adalah sebuah kurva $y(x)$ yang berbentuk ekponensial dari $x=1$ sampai $x=10$

4.3 Program Komputer Runge-Kutta orde -4 (multiple x)

4.3.1.Rancangan/ desain GUI



Gambar 4.7 Desain GUI – Runge-Kutta-4 (multiple x)

4.3.2 Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Pushbutton1	String	Display Graph
		Tag	show_graph_pushbutton
3	Static text1	String	nilai awal (x0,y0)
4	Static text2	String	nilai akhir (x)
5	Edit text1	String	
		Tag	Nilai_awal_x0_edit
6	Edit text2	String	
		Tag	Nilai_awal_y0_edit

7	Edit text3	String	
		Tag	Nilai_akhir_edit
8	Edit text4	String	
		Tag	nilai_akhir2_edit
9	Panel1	Title	ODE : Runge Kutta 4
12	Pushbutton1	String	calculate
		Tag	calculate_pushbutton

4.3.3. Kode Program (*source code*)

4.3.3.1 Function : *show_graph_pushbutton_Callback*

```
function show_graph_pushbutton_Callback(hObject, eventdata, handles)
axes(handles.axes1);
handles.x_0=0;
handles.x_interval=pi/100;
handles.x_n=2*pi;
handles.x=handles.x_0:handles.x_interval:handles.x_n;
handles.y=cos(handles.x);
plot(handles.x,handles.y);
title('Pers.Diff dy/dx = cos(x)');
xlabel('x');
ylabel('dy/dx');
hold on;
guidata(hObject,handles);
```

4.3.3.2 Function : *calculate_pushbutton_Callback*

```
x1 = str2num(get(handles.nilai_awal_x0_edit,'String'));
y1 = str2num(get(handles.nilai_awal_y0_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:handles.x_interval:x3; %x target from x2 to x3 with interval 1
for i=1:1:length(x_final)
    pita=100; %the number of segments / bands
    delx=(x2-x1)/pita; %the width of each segment /band
    xx=x1;
    yy_it(i)=y1;
    while (xx<x_final(i))
        xi=xx;
        yi=yy_it(i);
```

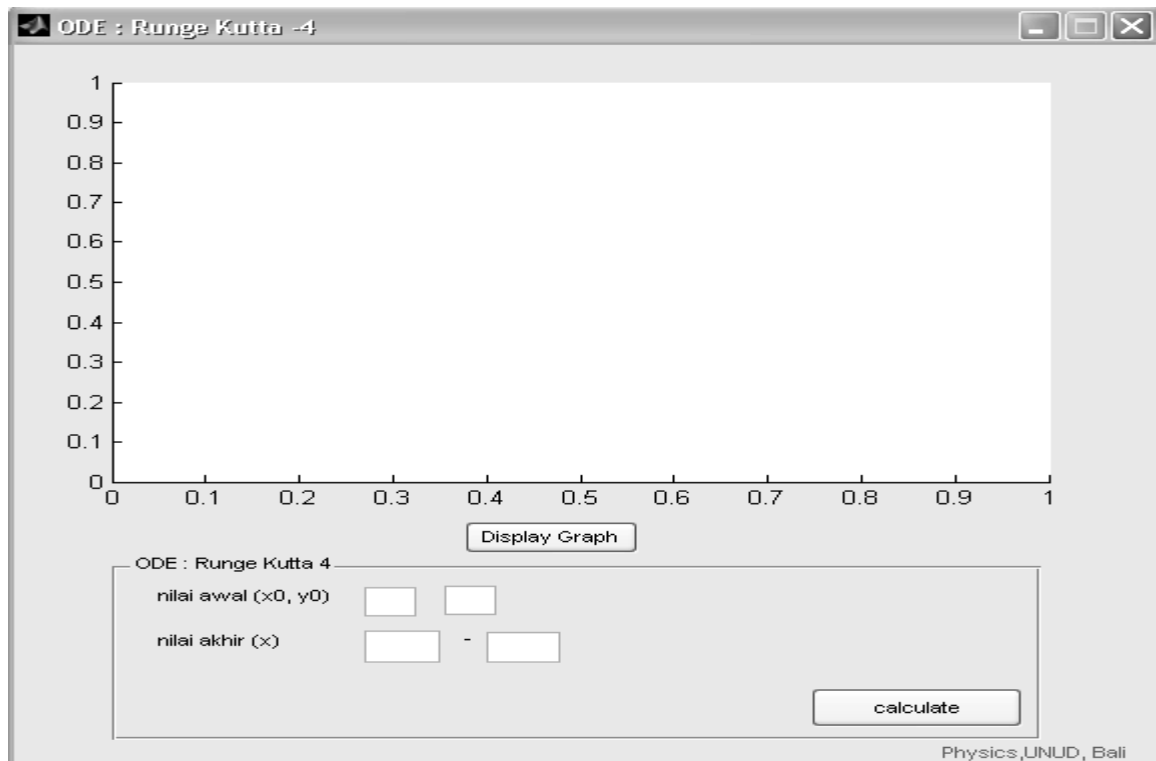
```

for j=1:1:4
    %Pers.Diferensial yang dicari solusinya
    %*****
    %fx=yi;
    fx=cos(xi);
    %*****
    k(j)=delx*fx;
    if j==1 || j==2
        xi=xx+delx/2;
        yi=yy_it(i)+k(j)/2;
    else
        if j==3
            xi=xx+delx;
            yi=yy_it(i)+k(j);
        end
    end
end
yy_it(i)=yy_it(i)+(k(1)+2*k(2)+2*k(3)+k(4))/6;%iterasi RK-4
xx=xx+delx;
end
end
cla;
axes(handles.axes1);
%display dy/dx
plot(handles.x,handles.y);
%text(10,10,'y');
title('Pers.Diff dy/dx = 4x+5');
xlabel('x');
ylabel('dy/dx');
%display y, the solution of dy/dx
plot(x_final,yy_it,'r');
legend('dy/dx','y');
hold on;

```

4.3.4 Hasil Eksekusi (*RUN*) program

Hasil eksekusi program Runge Kutta 4 (multiple x) dapat dilihat dalam gambar 4.8.

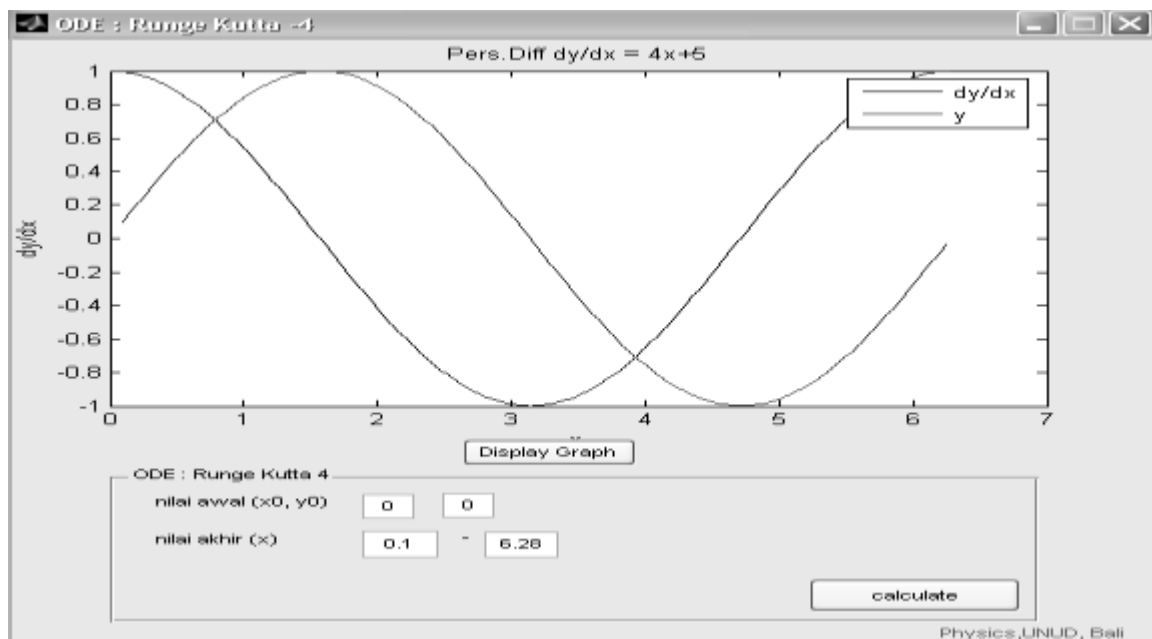


Gambar 4.8 Hasil Eksekusi program Runge Kutta 4 (multiple x)

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal (x0,y0) misal : 0 0
- Masukkan nilai akhir (x) misal : 0.1 6.28
- Klik tombol *calculate*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.9.



Gambar 4.9 Hasil Akhir eksekusi Runge Kutta 4 (multiple x)

Penjelasan program :

-persamaan diferensial yang akan dicari solusinya adalah :

$$dy/dx = \cos(x)$$

-nilai awal (x0,y0) : 0 0

Artinya x0 = 0 dan y0 = 0

-nilai akhir (x) : 0.1 6.28

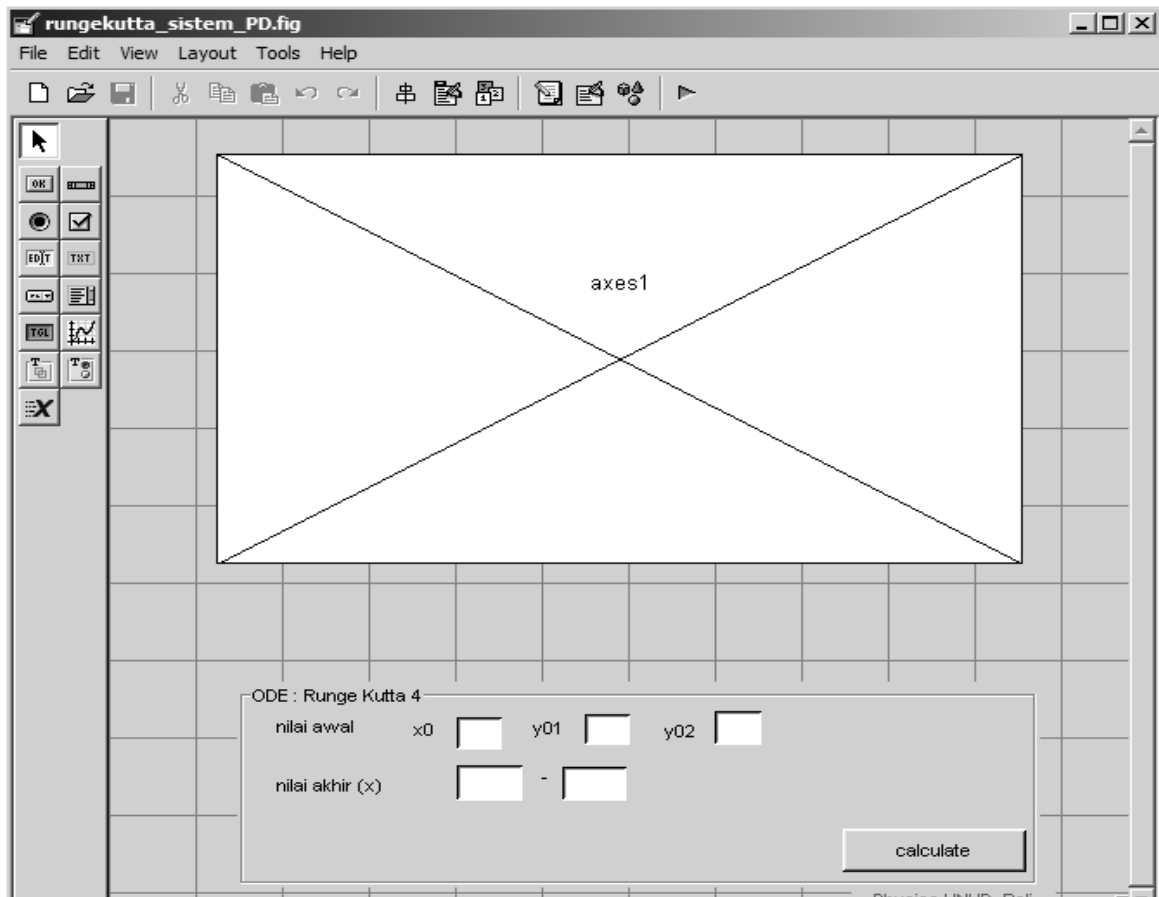
Artinya akan dihitung solusi persamaan diferensial saat x = 0.1 sampai x=6.28 atau y(0.1) sampai y(6.28)

- hasil solusinya adalah 2 buah kurva yaitu kurva $dy/dx=\cos(x)$ dan kurva solusinya y dari x=0.1 sampai x=6.28. Dapat dilihat solusinya y berupa kurva berbentuk fungsi $\sin(x)$

4.4 Solusi PD orde 2 untuk kasus Osilasi harmonik pada pegas

$$F = -kx ; \frac{d^2x}{dt^2} = -\frac{k}{m}x$$

4.4.1.Rancangan/ desain GUI



Gambar 4.10 Desain GUI – Pers. Diferensial orde 2 dengan Runge-Kutta-4

4.3.2 Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Static text1	String	nilai awal
3	Static text2	String	x0
4	Static text3	String	y01
5	Static text4	String	y02
6	Static text2	String	nilai akhir (x)
7	Edit text1	String	
		Tag	nilai_awal_x0_edit
8	Edit text2	String	
		Tag	nilai_awal_y01_edit
9	Edit text2	String	
		Tag	nilai_awal_y02_edit
10	Edit text3	String	
		Tag	nilai_akhir_edit
11	Edit text4	String	
		Tag	nilai_akhir2_edit
12	Panel1	Title	ODE : Runge Kutta 4
13	Pushbutton1	String	calculate
		Tag	calculate_pushbutton

4.4.3. Kode Program (*source code*)4.4.3.1 Function : *calculate_pushbutton_Callback*

```

x1 = str2num(get(handles.nilai_awal_x0_edit,'String'));
y1_1 = str2num(get(handles.nilai_awal_y01_edit,'String'));
y1_2 = str2num(get(handles.nilai_awal_y02_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:0.1:x3; %x target from x2 to x3 with interval
pita=1000; %the number of segments / bands
delx=(x2-x1)/pita; %the width of each segment /band
for i=1:1:length(x_final)
    %xx=x1;
    if i==1
        yy_it(i,1)=y1_1;
        yy_it(i,2)=y1_2;
        xx=x1;
    else

```

```

yy_it(i,1)=yy_it(i-1,1);
yy_it(i,2)=yy_it(i-1,2);
xx=xx+delx;
end
while (xx<x_final(i))
    xi=xx;
    y(1)=yy_it(i,1);
    y(2)=yy_it(i,2);
    for j=1:1:4
        %Pers.Diferensial yang dicari solusinya
        %*****
        k=100; %kasus pegas  $d^2x/dt^2=-k/m x$ 
        m=1;
        fx(1)=y(2);
        fx(2)=-k/m*y(1);
        %*****
        k(j,1)=delx*fx(1);
        k(j,2)=delx*fx(2);
        if j==1 || j==2
            xi=xx+delx/2;
            y(1)=yy_it(i,1)+k(j,1)/2;
            y(2)=yy_it(i,2)+k(j,2)/2;
        else
            if j==3
                xi=xx+delx;
                y(1)=yy_it(i,1)+k(j,1);
                y(2)=yy_it(i,2)+k(j,2);
            end
        end
    end
    yy_it(i,1)=yy_it(i,1)+(k(1,1)+2*k(2,1)+2*k(3,1)+k(4,1))/6; %iterasi RK-4
    yy_it(i,2)=yy_it(i,2)+(k(1,2)+2*k(2,2)+2*k(3,2)+k(4,2))/6;%iterasi RK-4
    xx=xx+delx;
end
end
cla;
axes(handles.axes1);
%display y, the solution of  $d^2y/dx^2$ 
plot(x_final,yy_it(:,1),'b');
title('Solusi Pers.Diff :  $d^2y/dx^2 = -k/m * x$ ');
xlabel('x');

```

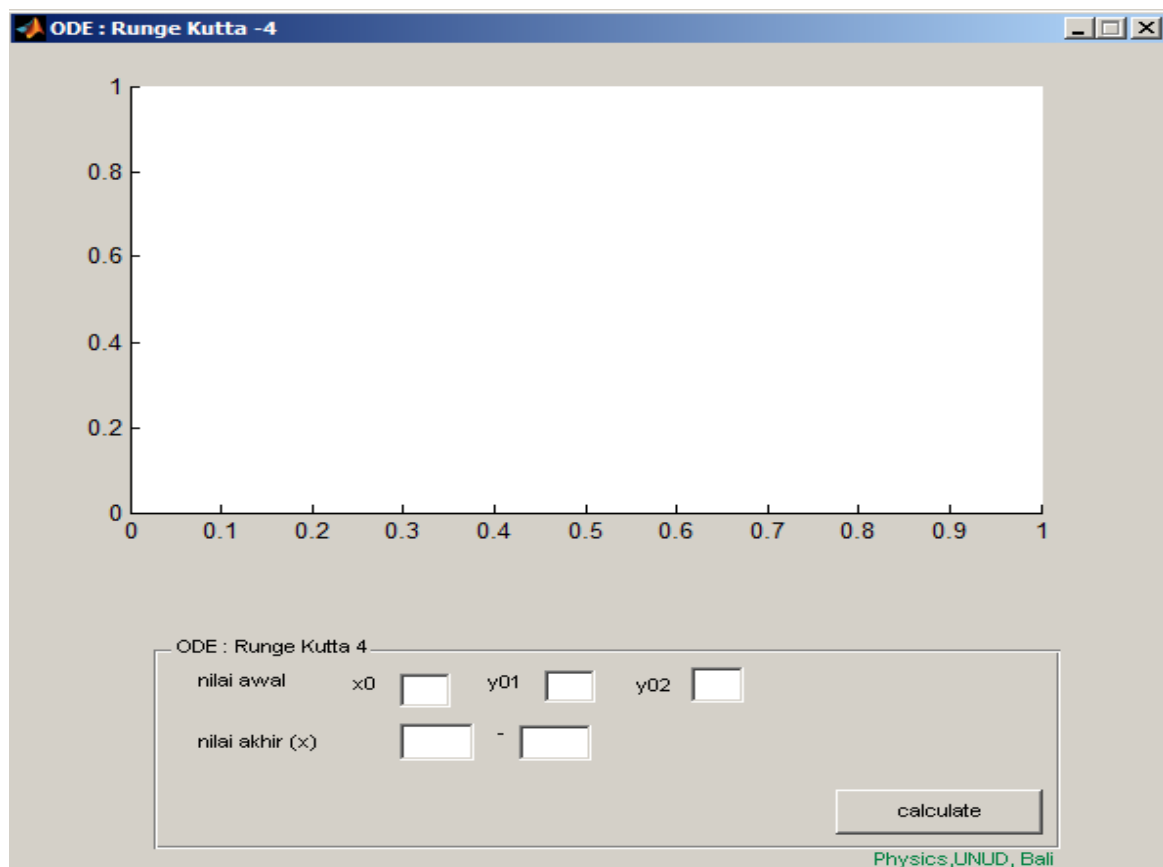
```

ylabel('y');
hold on;
plot(x_final,yy_it(:,2),'r');
legend('y','dy/dx');
hold on;

```

4.4.4 Hasil Eksekusi (*RUN*) program

Hasil eksekusi program Runge Kutta 4 untuk Persamaan Diferensial orde 2 dapat dilihat dalam gambar 4.11.



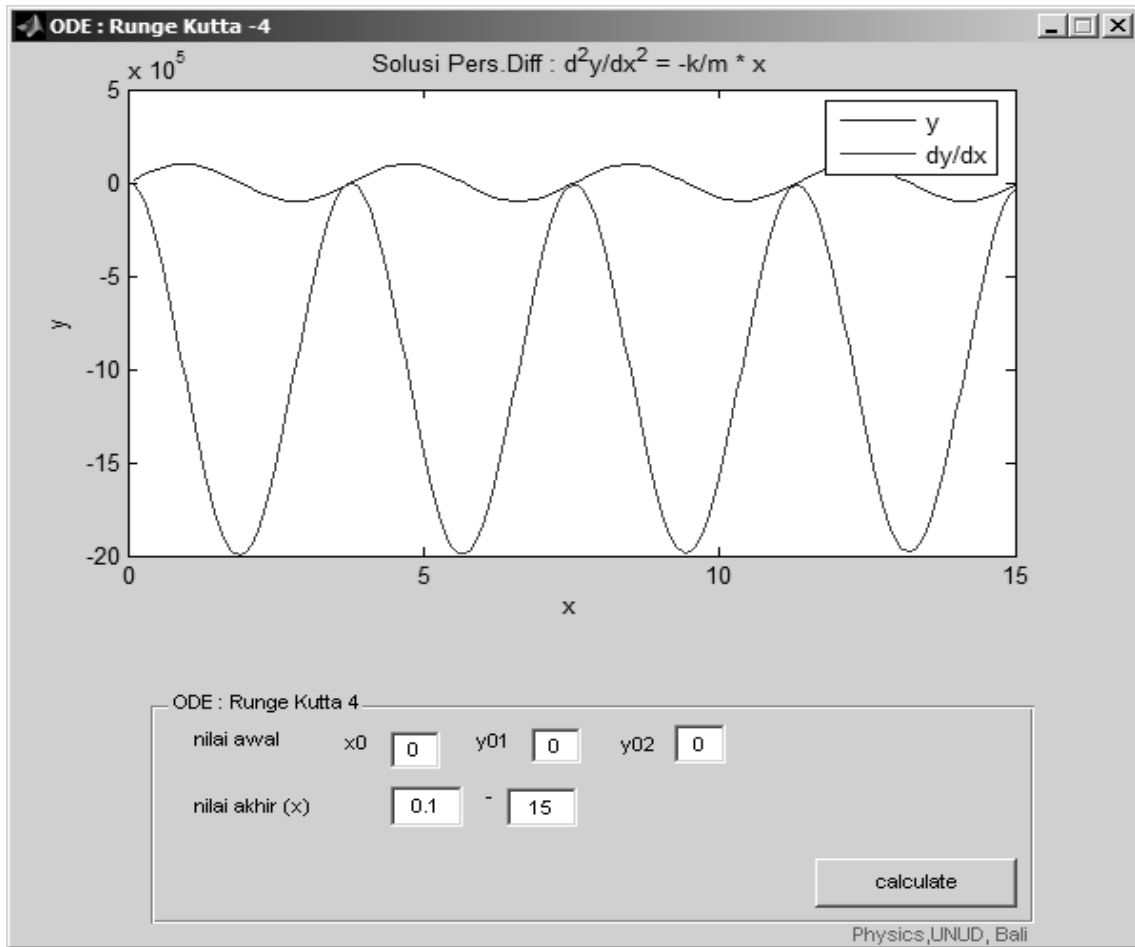
Gambar 4.11 Hasil Eksekusi program Solusi Pers. Diferensial orde 2 dengan Runge Kutta 4

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal x_0 , misal : 0
 y_{01} , misal : 0
 y_{02} , misal : 0
- Masukkan nilai akhir (x) misal : 0.1 15

- Klik tombol *calculate*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.12.



Gambar 4.12 Hasil Akhir program Solusi Pers. Diferensial orde 2 dengan metode Runge Kutta 4

Penjelasan program :

- persamaan diferensial orde 2 yang akan dicari solusinya adalah kasus pada pegas hukum Hooke:

$$F = -kx \quad \frac{d^2x}{dt^2} = -\frac{k}{m}x$$

Persamaan ini dijabarkan menjadi 2 buah persamaan diferensial orde 1 yaitu :

$$\frac{dx}{dt} = v \quad \frac{dv}{dt} = -\frac{k}{m}x$$

Jika persamaan diferensial diatas dinyatakan dalam f(x,y) maka :

$$\frac{d^2y}{dx^2} = -\frac{k}{m}y$$

Persamaan diferensial diatas dapat dituliskan dalam bentuk :

$$\frac{d^2 y_1}{dx^2} = -\frac{k}{m} y_1$$

Persamaan ini dapat dijabarkan menjadi 2 buah PD orde 1 yaitu :

$$\frac{dy_1}{dx} = y_2 \qquad \frac{dy_2}{dx} = -\frac{k}{m} y_1$$

Selanjutnya solusi dari kedua buah persamaan ini dicari dengan metode Runge-Kutta orde 4.

-nilai awal $x_0 : 0$, $y_{01} = 0$, $y_{02} = 0$

Artinya $x_0 = 0$, $y_{01} = 0$, dan $y_{02} = 0$

-nilai akhir (x) : 0.1 15

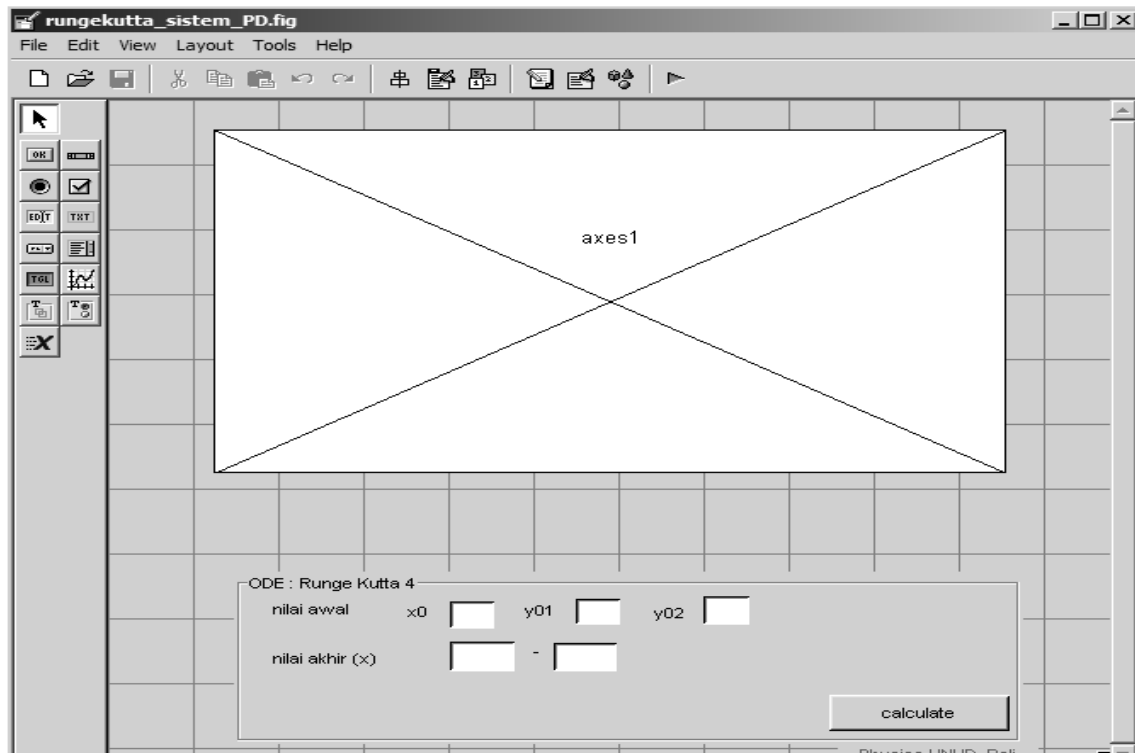
Artinya akan dihitung solusi persamaan diferensial saat $x = 0.1$ sampai $x=15$ atau $y(0.1)$ sampai $y(15)$

- hasil solusinya adalah 2 buah kurva yaitu kurva y dan kurva dy/dx dari $x=0.1$ sampai $x=15$. Dapat dilihat solusinya berupa kurva berbentuk fungsi $\sin(x)$

4.5 Solusi Persamaan Diferensial orde dua – kasus Osilasi teredam pada pegas

$$F = -kx - v \frac{dx}{dt} \qquad ; \qquad \frac{d^2 x}{dt^2} = -\frac{k}{m} x - v \frac{dx}{dt}$$

4.5.1.Rancangan/ desain GUI



Gambar 4.13 Desain GUI – PD orde 2 kasus osilasi teredam

4.5.2 Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Static text1	String	nilai awal
3	Static text2	String	x0
4	Static text3	String	y01
5	Static text4	String	y02
6	Static text2	String	nilai akhir (x)
7	Edit text1	String	
		Tag	Nilai_awal_x0_edit
8	Edit text2	String	
		Tag	Nilai_awal_y01_edit
9	Edit text2	String	
		Tag	Nilai_awal_y02_edit
10	Edit text3	String	
		Tag	Nilai_akhir_edit
11	Edit text4	String	
		Tag	nilai_akhir2_edit
12	Panel1	Title	ODE : Runge Kutta 4
13	Pushbutton1	String	calculate
		Tag	calculate_pushbutton

4.5.3. Kode Program (*source code*)

4.5.3.1 Function : *calculate_pushbutton_Callback*

```

x1 = str2num(get(handles.nilai_awal_x0_edit,'String'));
y1_1 = str2num(get(handles.nilai_awal_y01_edit,'String'));
y1_2 = str2num(get(handles.nilai_awal_y02_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:0.1:x3; %x target from x2 to x3 with interval
pita=1000; %the number of segments / bands
delx=(x2-x1)/pita; %the width of each segment /band
for i=1:1:length(x_final)
    %xx=x1;
    if i==1
        yy_it(i,1)=y1_1;
        yy_it(i,2)=y1_2;
        xx=x1;
    else
        yy_it(i,1)=yy_it(i-1,1);

```

```

yy_it(i,2)=yy_it(i-1,2);
xx=xx+delx;
end
while (xx<x_final(i))
    xi=xx;
    y(1)=yy_it(i,1);
    y(2)=yy_it(i,2);
    for j=1:1:4
        %Pers.Diferensial yang dicari solusinya
        %*****
        %kasus osilator harmonik teredam :  $d^2x/dt^2=-(k/m)x-\nu dx/dt$ 
        nu=2; %koefisien redaman
        k=100; %konstanta pegas
        m=1; %massa benda
        fx(1)=y(2); %dx/dt = v
        fx(2)= -(k/m)*y(1)-nu*y(2); %dv/dt = -(k/m)x-nu dx/dt
        %*****
        k(j,1)=delx*fx(1);
        k(j,2)=delx*fx(2);
        if j==1 || j==2
            xi=xx+delx/2;
            y(1)=yy_it(i,1)+k(j,1)/2;
            y(2)=yy_it(i,2)+k(j,2)/2;
        else
            if j==3
                xi=xx+delx;
                y(1)=yy_it(i,1)+k(j,1);
                y(2)=yy_it(i,2)+k(j,2);
            end
        end
    end
    yy_it(i,1)=yy_it(i,1)+(k(1,1)+2*k(2,1)+2*k(3,1)+k(4,1))/6; %iterasi RK-4
    yy_it(i,2)=yy_it(i,2)+(k(1,2)+2*k(2,2)+2*k(3,2)+k(4,2))/6;%iterasi RK-4
    xx=xx+delx;
end
end
end
cla;
axes(handles.axes1);
%display y, the solution of  $d^2y/dx^2$ 
plot(x_final,yy_it(:,1),'b');

```

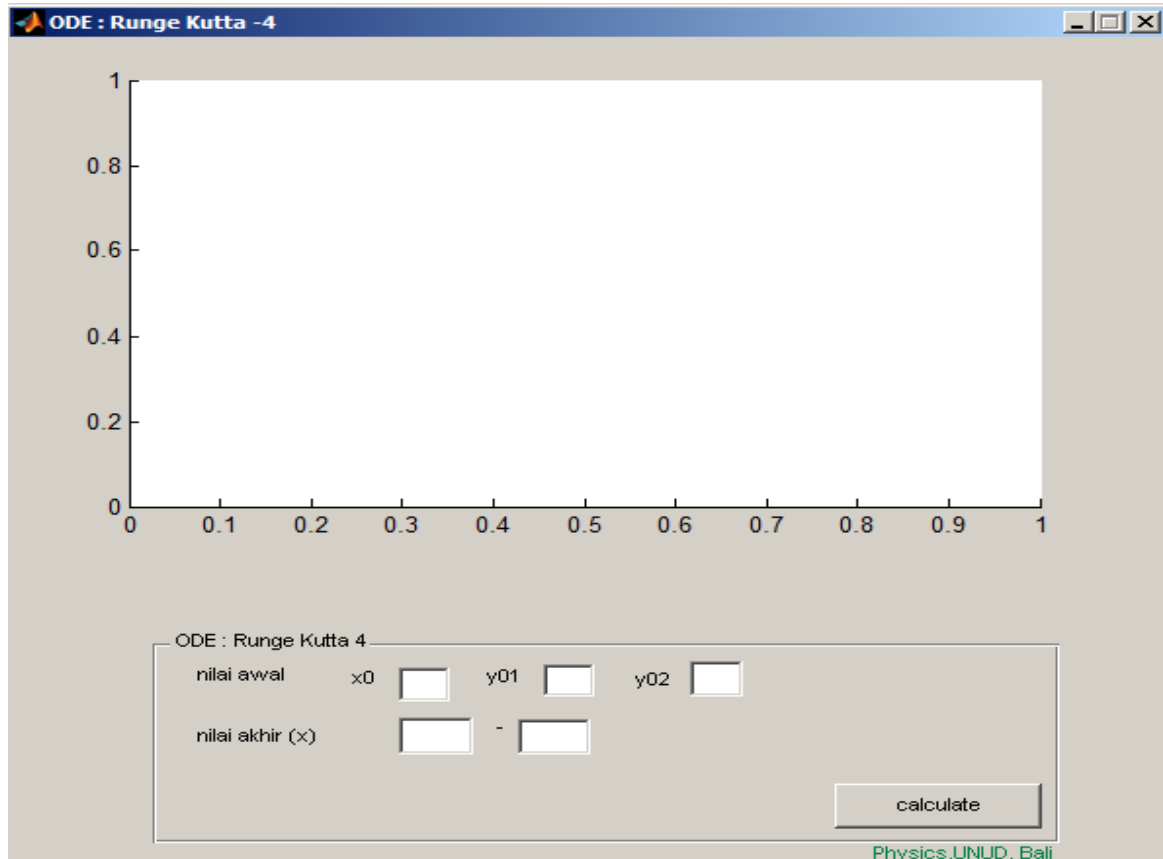
```

title('Solusi Pers.Diff : d^2y/dx^2 = -(k/m)*y-nu*dy/dx');
xlabel('x');
ylabel('y');
hold on;

```

4.5.4 Hasil Eksekusi (*RUN*) program

Hasil eksekusi program Runge Kutta 4 untuk Persamaan Diferensial orde 2 untuk kasus osilasi teredam dapat dilihat dalam gambar 4.14.

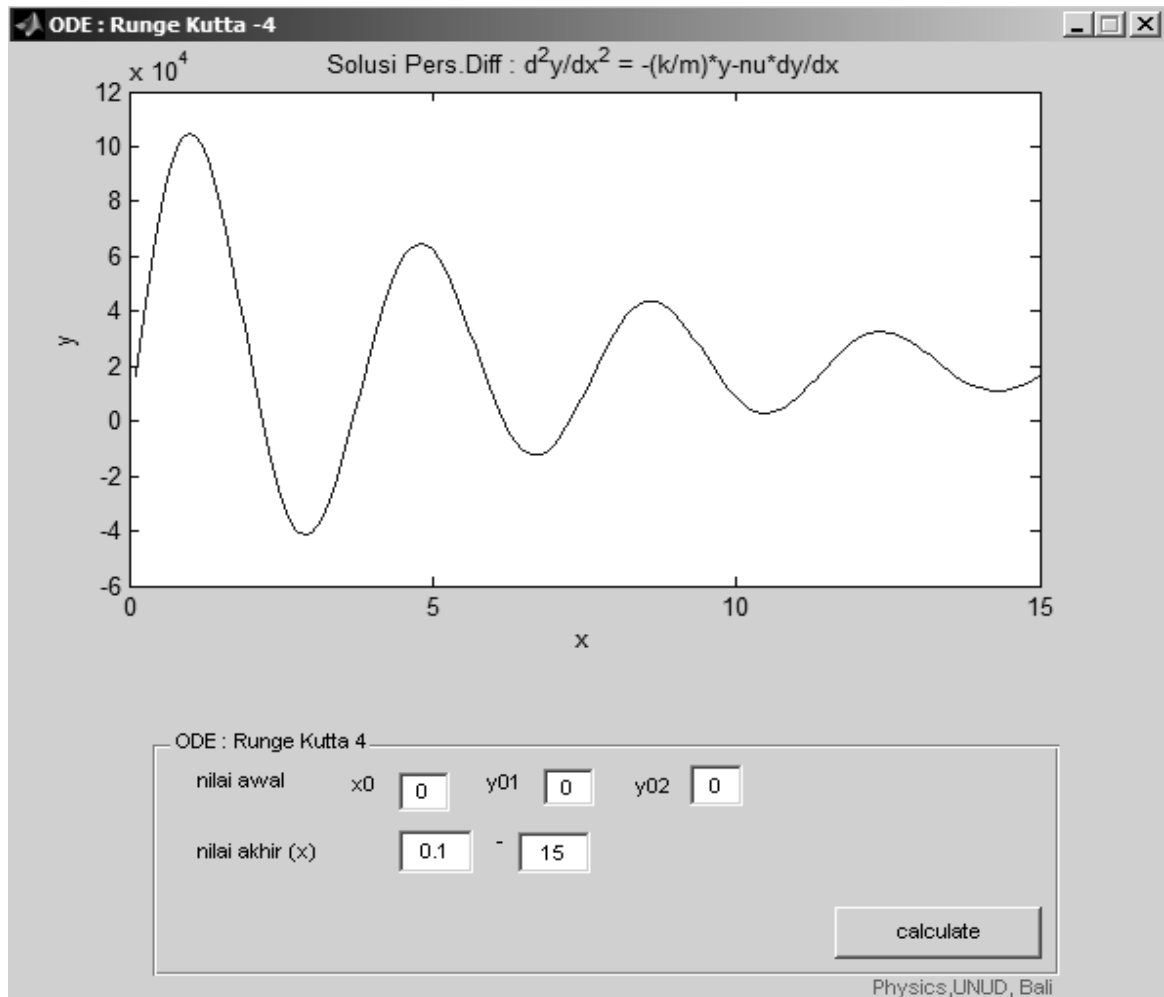


Gambar 4.14 Hasil Eksekusi program PD orde 2 kasus osilasi teredam pada pegas

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal x_0 ,misal : 0
 y_{01} , misal : 0
 y_{02} , misal : 0
- Masukkan nilai akhir (x) misal : 0.1 15
- Klik tombol *calculate*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.15.



Gambar 4.15 Hasil Akhir program Solusi PD orde 2– kasus osilasi teredam pegas

Penjelasan program :

- persamaan diferensial orde 2 yang akan dicari solusinya adalah kasus pada pegas hukum Hooke:

$$F = -kx - \nu \frac{dx}{dt} \quad \frac{d^2x}{dt^2} = -\frac{k}{m}x - \nu \frac{dx}{dt}$$

Persamaan ini dijabarkan menjadi 2 buah persamaan diferensial orde 1 yaitu :

$$\frac{dx}{dt} = v \quad \frac{dv}{dt} = -\frac{k}{m}x - \nu \frac{dx}{dt}$$

Jika persamaan diferensial diatas dinyatakan dalam f(x,y) maka :

$$\frac{d^2y}{dx^2} = -\frac{k}{m}y - \nu \frac{dy}{dx}$$

Persamaan diferensial diatas dituliskan :

$$\frac{d^2y_1}{dx^2} = -\frac{k}{m}y_1 - \nu \frac{dy_1}{dx}$$

Persamaan ini dijabarkan menjadi 2 buah persamaan diferensial orde 1 yaitu :

$$\frac{dy_1}{dx} = y_2 \qquad \frac{dy_2}{dx} = -\frac{k}{m}y_1 - \nu y_2$$

Selanjutnya solusi dari persamaan ini dicari dengan metode Runge-Kutta orde 4.

-nilai awal $x_0 : 0$, $y_{01} = 0$, $y_{02} = 0$

Artinya $x_0 = 0$, $y_{01} = 0$, dan $y_{02} = 0$

-nilai akhir (x) : 0.1 15

Artinya akan dihitung solusi persamaan diferensial saat $x = 0.1$ sampai $x=15$ atau $y(0.1)$ sampai $y(15)$

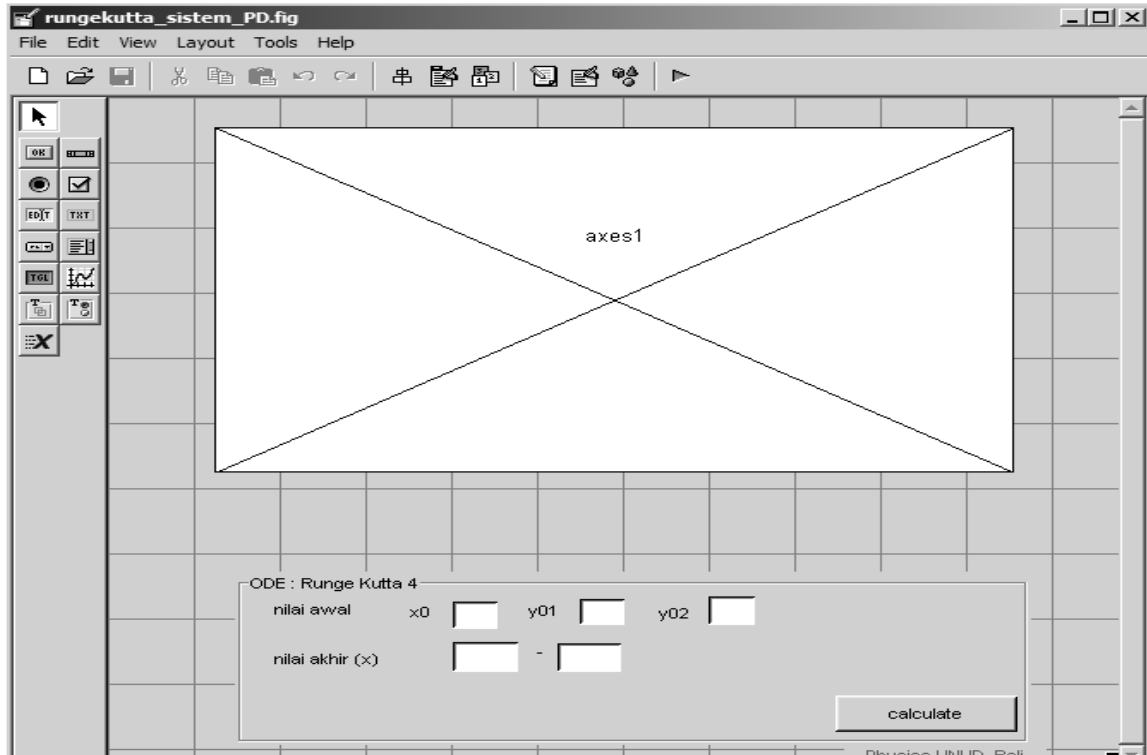
- hasil solusinya adalah sebuah kurva yaitu kurva y dari $x=0.1$ sampai $x=15$. Dapat dilihat solusinya berupa kurva berbentuk fungsi $\sin(x)$ dimana amplitudonya semakin kecil dengan bertambahnya nilai x .

4.6 Solusi PD orde 2 – kasus Osilasi teredam pada pegas yang ditambahkan gaya luar

$$F = -kx - \nu \frac{dx}{dt} + F(t) \qquad ; \qquad \frac{d^2x}{dt^2} = -\frac{k}{m}x - \nu \frac{dx}{dt} + F(t)$$

Dimana, gaya luar $F(t) = B \cos(\omega t)$

4.6.1.Rancangan/ desain GUI



Gambar 4.16 Desain GUI – PD orde 2 kasus osilasi teredam

4.6.2 Daftar Komponen

No	Komponen	Properti	
1	axes1	String	axes1
2	Static text1	String	nilai awal
3	Static text2	String	x0
4	Static text3	String	y01
5	Static text4	String	y02
6	Static text2	String	nilai akhir (x)
7	Edit text1	String	
		Tag	Nilai_awal_x0_edit
8	Edit text2	String	
		Tag	Nilai_awal_y01_edit
9	Edit text2	String	
		Tag	Nilai_awal_y02_edit
10	Edit text3	String	
		Tag	Nilai_akhir_edit
11	Edit text4	String	
		Tag	nilai_akhir2_edit
12	Panel1	Title	ODE : Runge Kutta 4
13	Pushbutton1	String	calculate
		Tag	calculate_pushbutton

4.6.3. Kode Program (*source code*)

4.6.3.1 Function : *calculate_pushbutton_Callback*

```

x1 = str2num(get(handles.nilai_awal_x0_edit,'String'));
y1_1 = str2num(get(handles.nilai_awal_y01_edit,'String'));
y1_2 = str2num(get(handles.nilai_awal_y02_edit,'String'));
%x from x2 to x3, y calculated for x from x2 to x3
x2 = str2num(get(handles.nilai_akhir_edit,'String'));
x3 = str2num(get(handles.nilai_akhir2_edit,'String'));
x_final = x2:0.1:x3; %x target from x2 to x3 with interval
pita=1000; %the number of segments / bands
delx=(x2-x1)/pita; %the width of each segment /band
for i=1:1:length(x_final)
    %xx=x1;
    if i==1
        yy_it(i,1)=y1_1;
        yy_it(i,2)=y1_2;
        xx=x1;
    else
        yy_it(i,1)=yy_it(i-1,1);

```

```

yy_it(i,2)=yy_it(i-1,2);
xx=xx+delx;
end
while (xx<x_final(i))
    xi=xx;
    y(1)=yy_it(i,1);
    y(2)=yy_it(i,2);
    for j=1:1:4
        %Pers.Differensial yang dicari solusinya
        %*****
        %kasus osilator harmonik teredam ditambah gaya luar:d^2x/dt^2=-(k/m)x-nu
dx/dt+F
        nu=2; %koefisien redaman
        k=100; %konstanta pegas
        m=1; %massa benda
        B=25e+5; %B amplitudo gaya luar, w frekuensi gaya luar
        w=100; %gaya luar periodik F = B cos(wt);
        fx(1)=y(2); %dx/dt = v
        fx(2)= -(k/m)*y(1)-nu*y(2)+B*cos((w*xi/57.3)); %dv/dt = -(k/m)x-nu dx/dt+B cos(wt)
        %*****
        k(j,1)=delx*fx(1);
        k(j,2)=delx*fx(2);
        if j==1 || j==2
            xi=xx+delx/2;
            y(1)=yy_it(i,1)+k(j,1)/2;
            y(2)=yy_it(i,2)+k(j,2)/2;
        else
            if j==3
                xi=xx+delx;
                y(1)=yy_it(i,1)+k(j,1);
                y(2)=yy_it(i,2)+k(j,2);
            end
        end
    end
    yy_it(i,1)=yy_it(i,1)+(k(1,1)+2*k(2,1)+2*k(3,1)+k(4,1))/6; %iterasi RK-4
    yy_it(i,2)=yy_it(i,2)+(k(1,2)+2*k(2,2)+2*k(3,2)+k(4,2))/6;%iterasi RK-4
    xx=xx+delx;
end
end
cla;
axes(handles.axes1);

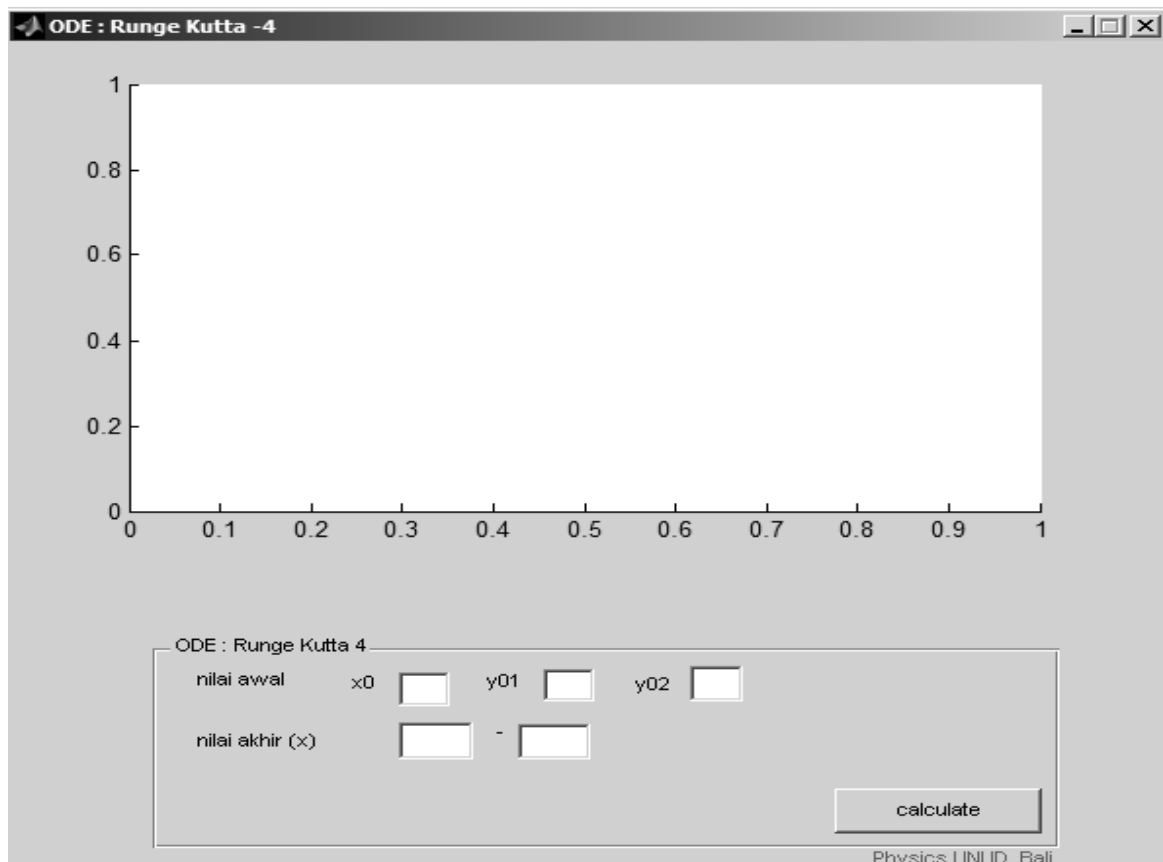
```



```
%display y, the solution of d^2y/dx^2
plot(x_final,yy_it(:,1),'b');
title('Solusi Pers.Diff : d^2y/dx^2 = -(k/m)*x-nu*dx/dt');
xlabel('x');
ylabel('y');
hold on;
```

4.6.4 Hasil Eksekusi (RUN) program

Hasil eksekusi program Runge Kutta 4 untuk Persamaan Diferensial orde 2 untuk kasus osilasi teredam ditambah gaya luar dapat dilihat dalam gambar 4.17.

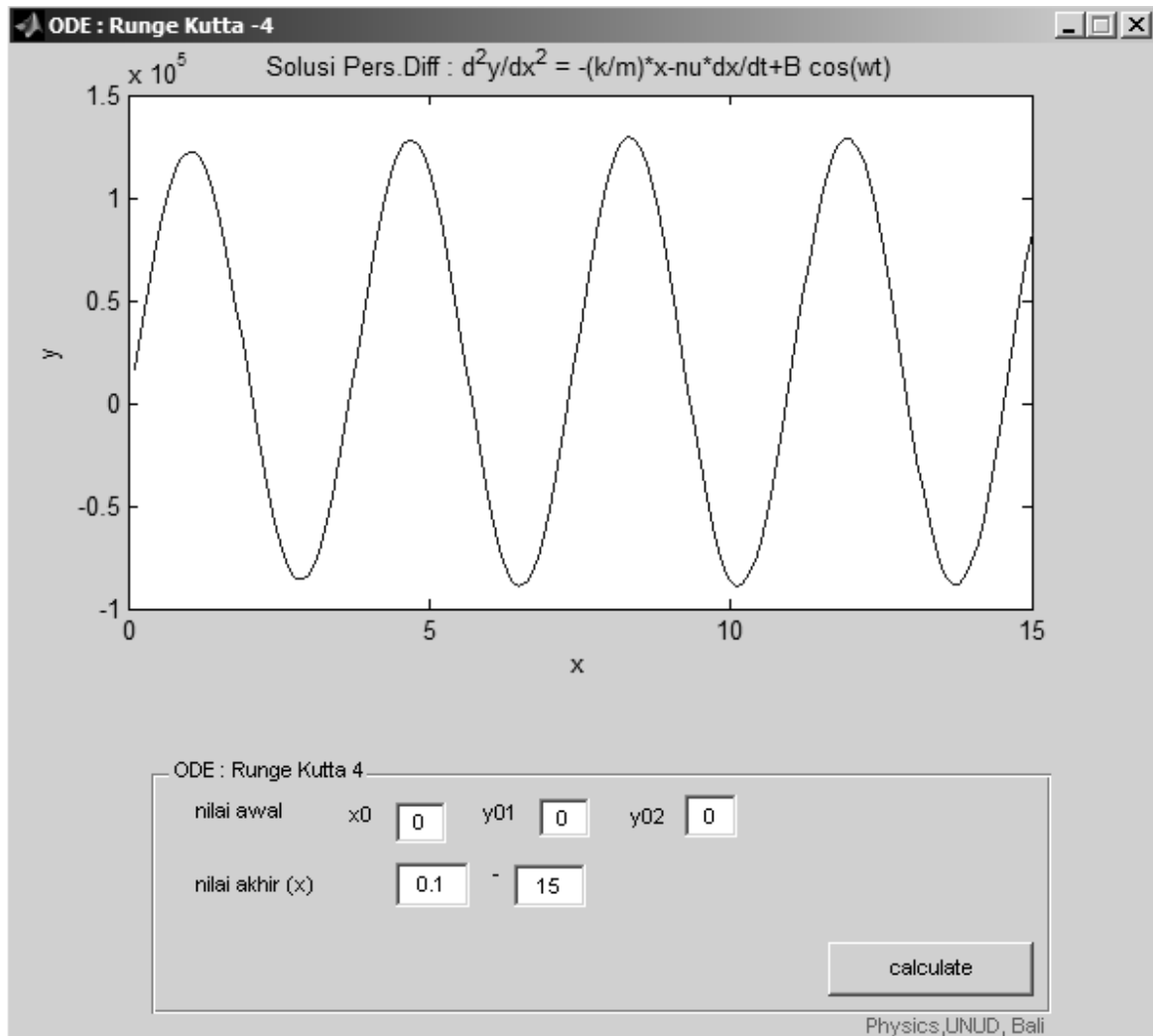


Gambar 4.17 Hasil Eksekusi program PD orde 2 kasus osilasi teredam pada pegas ditambah gaya luar

Cara kerja program :

- Klik tombol *show graph*
- Masukkan nilai awal x_0 ,misal : 0
 y_{01} , misal : 0
 y_{02} , misal : 0
- Masukkan nilai akhir (x) misal : 0.1 15
- Klik tombol *calculate*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 4.18.



Gambar 4.18 Hasil Solusi PD orde 2– kasus osilasi teredam pegas ditambah gaya luar

Penjelasan program :

- persamaan diferensial orde 2 yang akan dicari solusinya adalah kasus pada pegas teredam yang ditambahkan gaya luar $F(t)=B\cos(\omega t)$:

$$F = -kx - \nu \frac{dx}{dt} + F(t) \qquad \frac{d^2x}{dt^2} = -\frac{k}{m}x - \nu \frac{dx}{dt} + B\cos(\omega t)$$

Persamaan ini dijabarkan menjadi 2 buah persamaan diferensial orde 1 yaitu :

$$\frac{dx}{dt} = v \qquad \frac{dv}{dt} = -\frac{k}{m}x - \nu \frac{dx}{dt} + B\cos(\omega t)$$

Jika persamaan diferensial diatas dinyatakan dalam $f(x,y)$ maka :

$$\frac{d^2y}{dx^2} = -\frac{k}{m}y - \nu \frac{dy}{dx} + B\cos(\omega x)$$

PD diatas dituliskan menjadi :

$$\frac{d^2 y_1}{dx^2} = -\frac{k}{m} y_1 - \nu \frac{dy_1}{dx} + B \cos(\omega x)$$

Persamaan ini dijabarkan menjadi 2 buah persamaan diferensial orde 1 yaitu :

$$\frac{dy_1}{dx} = y_2 \qquad \frac{dy_2}{dx} = -\frac{k}{m} y_1 - \nu y_2 + B \cos(\omega x)$$

Selanjutnya solusi dari persamaan ini dicari dengan metode Runge-Kutta orde 4.

-nilai awal $x_0 : 0$, $y_{01} = 0$, $y_{02} = 0$

Artinya $x_0 = 0$, $y_{01} = 0$, dan $y_{02} = 0$

-nilai akhir (x) : 0.1 15

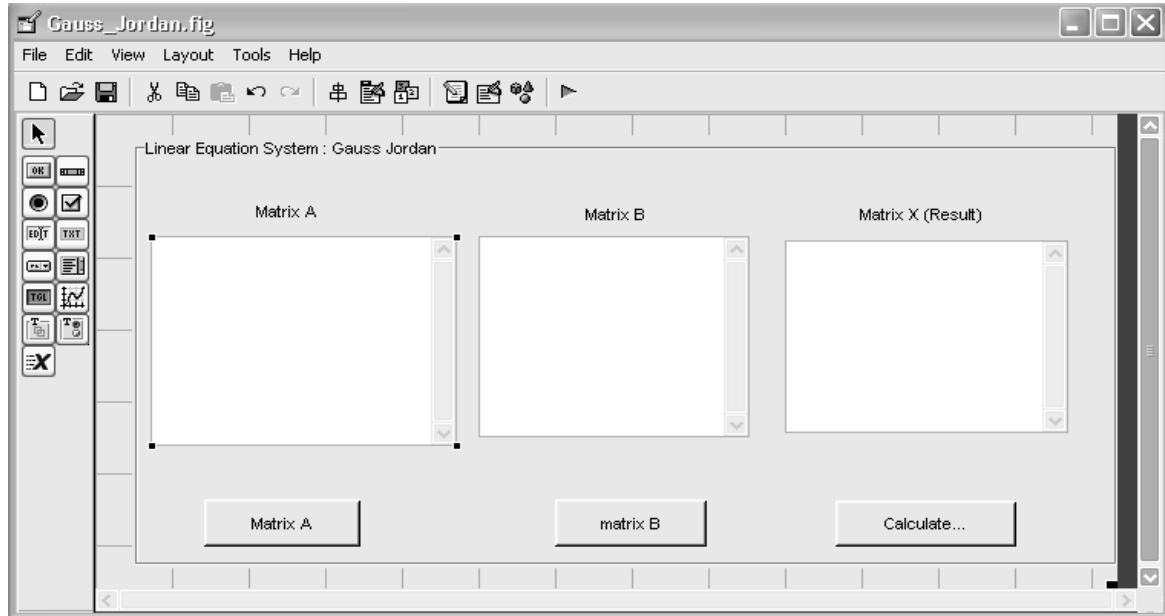
Artinya akan dihitung solusi persamaan diferensial saat $x = 0.1$ sampai $x=15$ atau $y(0.1)$ sampai $y(15)$

- hasil solusinya adalah sebuah kurva y dari $x=0.1$ sampai $x=15$. Dapat dilihat solusinya berupa kurva berbentuk fungsi $\sin(x)$ dimana amplitudonya konstan tidak terjadi peredaman sebagai akibat dari penambahan gaya luar yang bersifat periodik berbentuk fungsi cosinus yaitu $F(t) = B \cos(\omega t)$

V SISTEM PERSAMAAN LINEAR

Program Komputer Gauss – Jordan

1. Rancangan Desain



Gambar 3.1 Desain GUI – Gauss Jordan

2. Daftar Komponen

No	Komponen	Properti	
1	Listbox1	String	-
		Tag	matrix_A_listbox
2	Listbox2	String	-
		Tag	matrix_B_listbox
3	Listbox3	String	-
		Tag	matrix_x_listbox
4	Pushbutton1	String	matrix A
		Tag	Matrix_A_pushbutton
5	Pushbutton2	String	matrix B
		Tag	Matrix_B_pushbutton
6	Pushbutton3	String	Calculate...
		Tag	calculate_pushbutton
7	Panel1	Title	Linear Equation System : Gauss Jordan

8	Static text1	String	Matix A
9	Static text2	String	Matix B
10	Static text3	String	Matix C (result)

3. Kode Program (*source code*)

3.1.Function : *Matrix_A_pushbutton_Callback*

```
function Matrix_A_pushbutton_Callback(hObject, eventdata, handles)
[filename,handles.pathname]=uigetfile('*.xls;*.xlsx','Select data training');
handles.data_fileA = strcat(handles.pathname,filename);
if exist(handles.data_fileA) % check if files exists
    handles.dataA=xlsread(handles.data_fileA); % read data training in 'excels' file
end
set(handles.matrix_A_listbox,'String',num2str(handles.dataA));
guidata(hObject,handles);
```

3.2.Function : *Matrix_B_pushbutton_Callback*

```
[filename,handles.pathname]=uigetfile('*.xls;*.xlsx','Select data training');
handles.data_fileB = strcat(handles.pathname,filename);
if exist(handles.data_fileB) % check if files exists
    handles.dataB=xlsread(handles.data_fileB); % read data training in 'excels' file
end
set(handles.matrix_B_listbox,'String',num2str(handles.dataB));
guidata(hObject,handles);
```

3.3.Function : *calculate_pushbutton_Callback*

```
function calculate_pushbutton_Callback(hObject, eventdata, handles)
n=size(handles.dataA,1);
A=handles.dataA;
B=handles.dataB;
%calculate X from AX = B by using Gauss-Jordan elimination
AB=[A B']; %Augmented Matrix
X=rref(AB); %reduced row echelon form (eselon baris tereduksi)
XX=X(:,n+1);%The result are located on the last column of matrix X
set(handles.matrix_x_listbox,'String',num2str(XX));
```

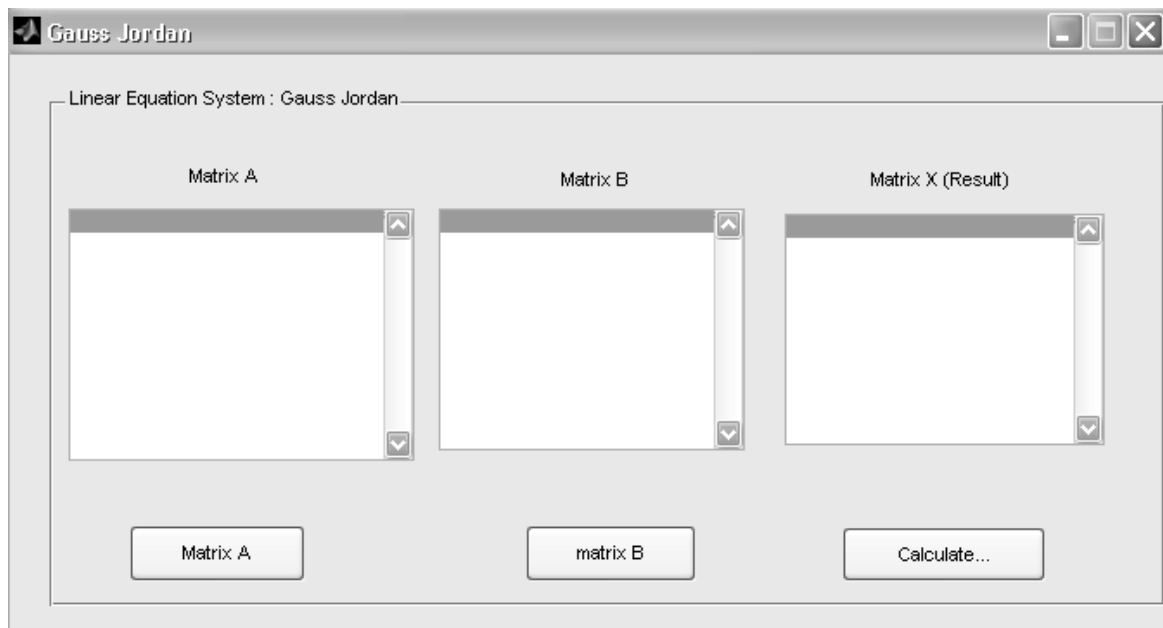
4. Buat 2 buah file Excel,

Untuk menyimpan matrik A dan matrik B maka diperlukan 2 buah file Excel yaitu :

- 1.matrikA.xlsx
- 2.matrikB.xlsx

5. Hasil Eksekusi (*RUN*) program

Hasil eksekusi program Gauss Jordan dapat dilihat dalam gambar 3.2.

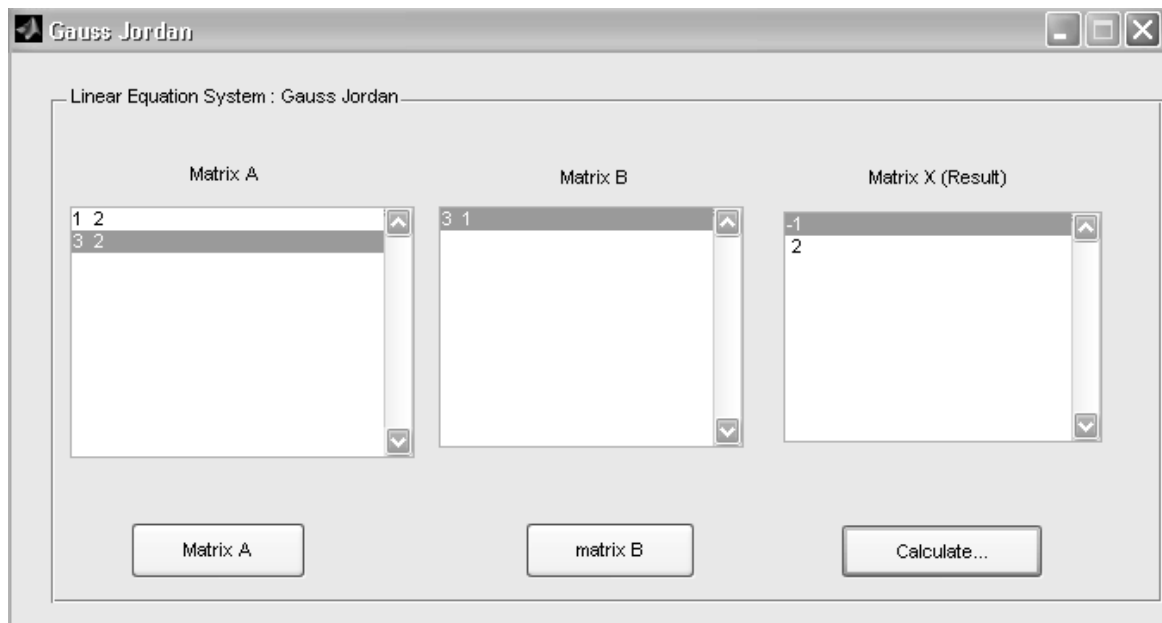


Gambar 3.2 Hasil Eksekusi program Gauss Jordan

Cara kerja program :

- Klik tombol *Matrik A*, selanjutnya pilih file Excel yaitu : matrikA.xlsx
- Klik tombol *Matrik B*, selanjutnya pilih file Excel yaitu : matrikB.xlsx
- Klik tombol *Calculate...*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 3.3.



Gambar 3.3 Hasil Akhir eksekusi program Gauss Jordan

Penjelasan program :

- matrik A adalah :

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$$

-matrik B adalah :

$$B = [3 \quad 1]$$

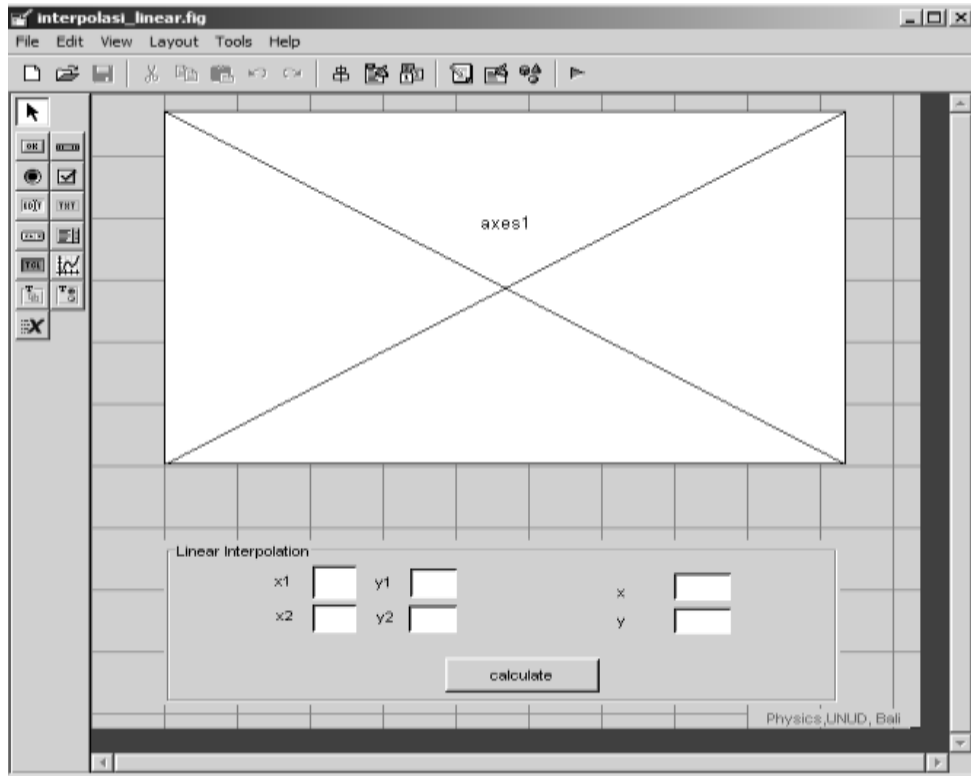
-Hasilnya adalah matrik X :

$$X = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

6 INTERPOLASI

6.1 Program Komputer Interpolasi Linier

6.1.1. Rancangan Desain



Gambar 6.1 Desain GUI : Interpolasi Linier

6.1.2 Daftar Komponen

No	Komponen	Properti	
1	Axes1	Tag	axes1
2	Static text1	String	x1
3	Static text2	String	x2
4	Static text3	String	y1
5	Static text4	String	y2
6	Static text5	String	x
7	Static text6	String	y
8	Edit Text1	String	
		Tag	x1_edit

9	Edit Text2	String	
		Tag	y1_edit
10	Edit Text3	String	
		Tag	x2_edit
11	Edit Text4	String	
		Tag	y2_edit
12	Edit Text5	String	
		Tag	x_edit
13	Edit Text6	String	
		Tag	y_edit
14	Pushbutton1	String	calculate
		Tag	calculate_pushbutton
15	Panel1	title	Linear Interpolation

6.1.3. Kode Program (*source code*)

6.1.3.1 Function : *calculate_pushbutton_Callback*

```
function calculate_pushbutton_Callback(hObject, eventdata, handles)
hold off;
x1 = str2num(get(handles.x1_edit,'String'));
y1 = str2num(get(handles.y1_edit,'String'));
x2 = str2num(get(handles.x2_edit,'String'));
y2 = str2num(get(handles.y2_edit,'String'));
x = str2num(get(handles.x_edit,'String'));
y=(((y2-y1)/(x2-x1))*(x-x1))+y1;
%plotting data
xx(1)=x1;
yy(1)=y1;
xx(2)=x2;
yy(2)=y2;
xx(3)=x;
yy(3)=y;
plot(xx,yy,'o');
hold on;
set(handles.y_edit,'String',num2str(y));
%plotting y
axes(handles.axes1);
x=0:0.1:x2+1;
```

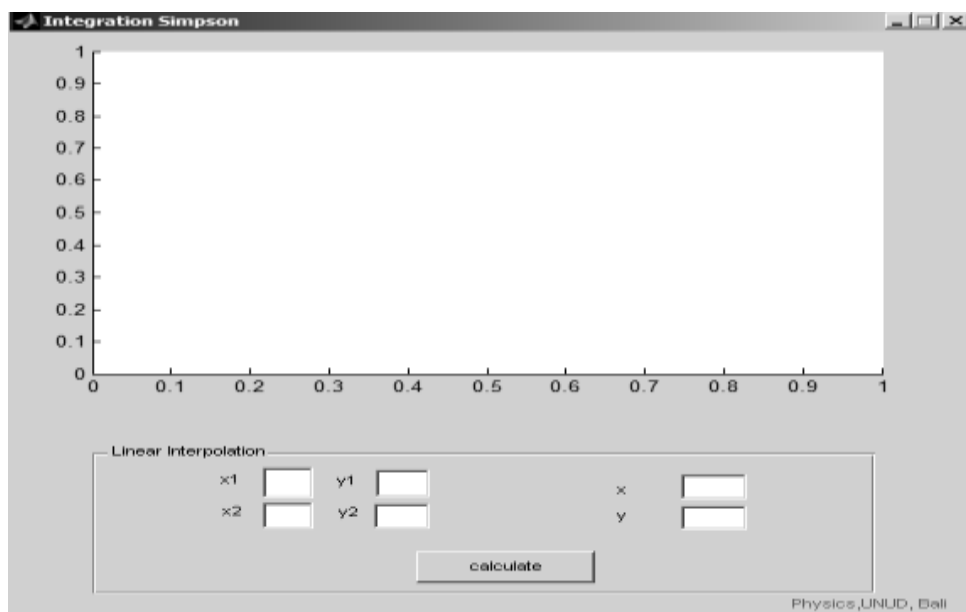
```

y=((y2-y1)/(x2-x1)).*(x-x1))+y1;
%x=0:10;
%y=x.^2;
plot(x,y);
title('y=((y2-y1)/(x2-x1)).*(x-x1))+y1');
xlabel('x');
ylabel('y');

```

6.1.4 Hasil Eksekusi (RUN) program

Hasil eksekusi program Interpolasi Linier dapat dilihat dalam gambar 6.2.

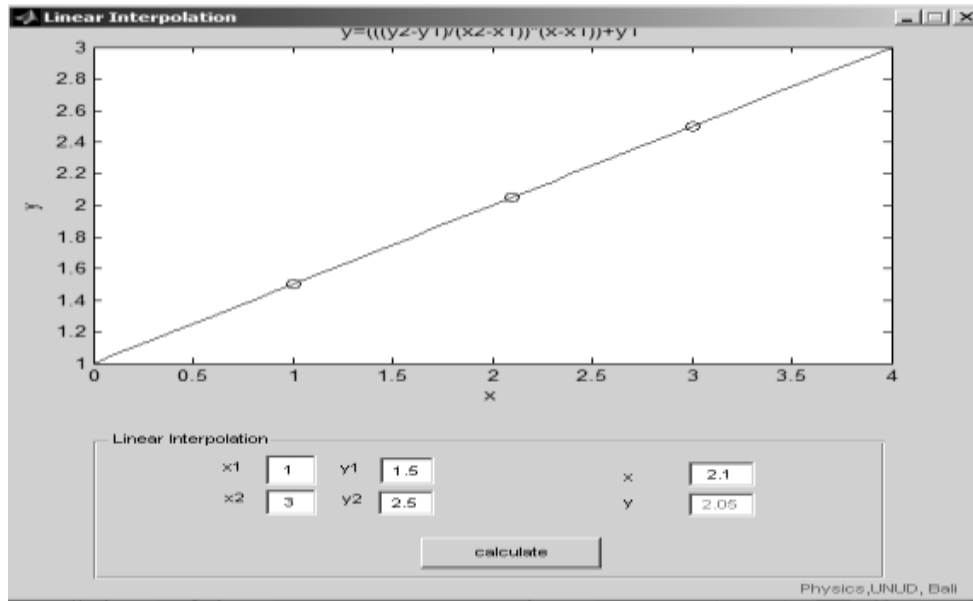


Gambar 6.2 Hasil Eksekusi program Interpolasi Linier

Cara kerja program :

- Masukkan x_1 y_1 , misal : 1 1.5
- Masukkan x_2 y_2 , misal : 3 2.5
- Masukkan x , misal : 2.1
- Klik tombol *calculate*

Maka akan didapatkan hasil seperti diperlihatkan dalam gambar 6.3.



Gambar 6.3 Hasil Akhir eksekusi program Interpolasi Linier
Penjelasan hasil program :

-masukkan $x1$ $y1$: 1 1.5

-masukkan $x2$ $y2$: 3 2.5

Memasukkan koordinat titik-titik yang akan dicari interpolasinya

-masukkan x : 2.1

Menentukan nilai absis dari titik yang akan dicari interpolasinya

-Hasilnya, y : 2.05

Hasil interpolasi linear dari titik-titik tersebut adalah 2.05

DAFTAR PUSTAKA

- Abdul Munif, Aries Prastyoko H., *Cara Praktis Penguasaan dan Penggunaan Metode Numerik*, Penerbit Guna Widya, Surabaya, 1995
- Ahmad Basuki, Nana Ramadijanti, *Metode Numerik dan Algoritma Komputasi*, Penerbit Andi, Jogjakarta, 2005
- Chapra Steven C., Canale Raymond P., *Numerical Methods for Engineers*, Mc Graw-Hill Book Company, New York, 1985
- De Vries, P.L., *A First Course in Computational Physics*, John Wiley & Sons, USA, 1999
- R.Soegeng, *Komputasi Numerik dengan Turbo Pascal*, Penerbit Andi, Jogjakarta, 1993
- Young, M.J. *Mastering Visual C++*, Sibex, 1998